

Description of the Audio Equalizer and a Sample Run with FIR & IIR Analysis

by

Adham Mohamed Aly — 6744

Yousef Ashraf Kotp — 7140

Mohamed Farid Abdelaziz — 6905

Group 1 Section 2

REPORT

Submitted in Partial Fulfillment of the
Requirements of the Final Project of

Digital Signal Processing
CC332 Course

Dr. Ahmed Said El-Tarras

Table of Contents

Code	1
Sample Run	20
FIR Analysis.....	20
IIR Analysis	35
Doubling the sample rate.....	49
Decreasing to half	51

Code

AudioEqualizer.mlapp (GUI)

```
classdef AudioEqualizer < matlab.apps.AppBase
```

```
% Properties that correspond to app components
properties (Access = public)
    UIFigure matlab.ui.Figure
    ProcessingLabel matlab.ui.control.Label
    ANALYZEButton matlab.ui.control.Button
    KHzKnob_5_Detail matlab.ui.control.Label
    KHzKnob_4_Detail matlab.ui.control.Label
    KHzKnob_3_Detail matlab.ui.control.Label
    KHzKnob_2_Detail matlab.ui.control.Label
    KHzKnob_1_Detail matlab.ui.control.Label
    HzKnob_4_Detail matlab.ui.control.Label
    HzKnob_3_Detail matlab.ui.control.Label
    HzKnob_2_Detail matlab.ui.control.Label
    HzKnob_1_Detail matlab.ui.control.Label
    FrequencyBands matlab.ui.control.Label
    SAVEButton matlab.ui.control.Button
    stopButton matlab.ui.control.Button
    playButton matlab.ui.control.Button
    RESETButton matlab.ui.control.Button
    APPLYFILTERSButton matlab.ui.control.Button
    KHzKnob_5 matlab.ui.control.Knob
    KHzKnob_5Label matlab.ui.control.Label
    KHzKnob_4 matlab.ui.control.Knob
    KHzKnob_4Label matlab.ui.control.Label
    KHzKnob_3 matlab.ui.control.Knob
    KHzKnob_3Label matlab.ui.control.Label
    KHzKnob_2 matlab.ui.control.Knob
    KHzKnob_2Label matlab.ui.control.Label
    KHzKnob_1 matlab.ui.control.Knob
```

```

KHzKnobLabel matlab.ui.control.Label
HzKnob_4 matlab.ui.control.Knob
HzKnob_11Label matlab.ui.control.Label
HzKnob_3 matlab.ui.control.Knob
HzLabel matlab.ui.control.Label
HzKnob_2 matlab.ui.control.Knob
HzKnob_10Label matlab.ui.control.Label
HzKnob_1 matlab.ui.control.Knob
HzKnobLabel matlab.ui.control.Label
sampleRateField matlab.ui.control.NumericEditField
sampleRateLabel matlab.ui.control.Label
SelectButton matlab.ui.control.Button
NOAUDIOFILECHOSENLabel matlab.ui.control.Label
FILTERTYPEButtonGroup matlab.ui.container.ButtonGroup
IIRButton matlab.ui.control.RadioButton
FIRButton matlab.ui.control.RadioButton
AUDIOEQUALIZERLabel matlab.ui.control.Label
filteredPlot matlab.ui.control.UIAxes
originalPlot matlab.ui.control.UIAxes

```

```

properties (Access = public)
originalSampleRate = 0 % SAMPLE RATE LOADED FROM AUDIO FILE
sampleRate = 0 % SAMPLE RATE USED IN PROCESSING
originalAudio % AUDIO DATA LOADED FROM AUDIO FILE
audio % AUDIO DATA USED IN PROCESSING (SR MAY DIFFER)
fileName
filePath
filterClass = 0 % 0 FOR FIR & 1 FOR IIR
enhancedAudio % AUDIO DATA AFTER FILTERING
frequencyBands = ones(1, 9)
end

```

```

% Callbacks that handle component events
methods (Access = private)

```

```

% Code that executes after component creation
function startupFcn(app)
clear;
end

```

```

% Button pushed function: SelectButton
function SelectButtonPushed(app, event)
[file, path] = uigetfile('*.wav', 'Select File');
if (file ~= 0)
app.fileName = file;
app.filePath = strcat(path, file);

```

```

app.ProcessingLabel.Visible = 'on';

[app.originalAudio, app.originalSampleRate] = audioread(app.filePath);

app.sampleRateField.Value = app.originalSampleRate;
app.sampleRateLabel.Text = strcat(' SAMPLE RATE:', {' '},
num2str(app.originalSampleRate));
app.sampleRate = app.originalSampleRate;

app.audio = app.originalAudio;
app.NOAUDIOFILECHOSENLabel.Text = strcat(' AUDIO FILE CHOSEN:', {' '},
app.fileName);

APPLYFILTERSButtonPushed(app, event);
FILTERTYPEButtonGroupSelectionChanged(app, event);

app.ProcessingLabel.Visible = 'off';
end
end

% Selection changed function: FILTERTYPEButtonGroup
function FILTERTYPEButtonGroupSelectionChanged(app, event)
selectedButton = app.FILTERTYPEButtonGroup.SelectedObject;
if(selectedButton == app.FIRButton)
app.filterClass = 0;
disp('using FIR');
else
app.filterClass = 1;
disp('using IIR');
end
end

% Value changed function: HzKnob_1
function HzKnob_1ValueChanged(app, event)
value = app.HzKnob_1.Value;
app.frequencyBands(1) = 10^(value/20);
app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
app.HzKnob_1_Detail.Text = mat2str(10^(value/20), 2);
end

% Value changed function: HzKnob_2
function HzKnob_2ValueChanged(app, event)
value = app.HzKnob_2.Value;
app.frequencyBands(2) = 10^(value/20);

```

```

app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
app.HzKnob_2_Detail.Text = mat2str(10^(value/20), 2);
end

```

```

% Value changed function: HzKnob_3
function HzKnob_3ValueChanged(app, event)
value = app.HzKnob_3.Value;
app.frequencyBands(3) = 10^(value/20);
app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
app.HzKnob_3_Detail.Text = mat2str(10^(value/20), 2);
end

```

```

% Value changed function: HzKnob_4
function HzKnob_4ValueChanged(app, event)
value = app.HzKnob_4.Value;
app.frequencyBands(4) = 10^(value/20);
app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
app.HzKnob_4_Detail.Text = mat2str(10^(value/20), 2);
end

```

```

% Value changed function: KHzKnob_1
function KHzKnob_1ValueChanged(app, event)
value = app.KHzKnob_1.Value;
app.frequencyBands(5) = 10^(value/20);
app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
app.KHzKnob_1_Detail.Text = mat2str(10^(value/20), 2);
end

```

```

% Value changed function: KHzKnob_2
function KHzKnob_2ValueChanged(app, event)
value = app.KHzKnob_2.Value;
app.frequencyBands(6) = 10^(value/20);
app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
app.KHzKnob_2_Detail.Text = mat2str(10^(value/20), 2);
end

```

```

% Value changed function: KHzKnob_3
function KHzKnob_3ValueChanged(app, event)
    value = app.KHzKnob_3.Value;
    app.frequencyBands(7) = 10^(value/20);
    app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
    app.KHzKnob_3_Detail.Text = mat2str(10^(value/20), 2);
end

```

```

% Value changed function: KHzKnob_4
function KHzKnob_4ValueChanged(app, event)
    value = app.KHzKnob_4.Value;
    app.frequencyBands(8) = 10^(value/20);

```

```

        app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
        app.KHzKnob_4_Detail.Text = mat2str(10^(value/20), 2);
    end

% Value changed function: KHzKnob_5
function KHzKnob_5ValueChanged(app, event)
    value = app.KHzKnob_5.Value;
    app.frequencyBands(9) = 10^(value/20);
    app.FrequencyBands.Text = mat2str(app.frequencyBands, 2);
    app.KHzKnob_5_Detail.Text = mat2str(10^(value/20), 2);
end

% Button pushed function: APPLYFILTERSButton
function APPLYFILTERSButtonPushed(app, event)
    app.ProcessingLabel.Visible = 'on';

    app.enhancedAudio = apply_filters(app.originalAudio,
app.originalSampleRate, app.frequencyBands, app.filterClass);

    plot(app.originalPlot, app.originalAudio);
    plot(app.filteredPlot, app.enhancedAudio);

    app.ProcessingLabel.Visible = 'off';
end

```

```

app.sampleRate = app.originalSampleRate;
app.frequencyBands = ones(1, 9);

app.sampleRateField.Value = app.originalSampleRate;
app.HzKnob_1.Value = 0;
app.HzKnob_2.Value = 0;
app.HzKnob_3.Value = 0;
app.HzKnob_4.Value = 0;
app.KHzKnob_1.Value = 0;
app.KHzKnob_2.Value = 0;
app.KHzKnob_3.Value = 0;
app.KHzKnob_4.Value = 0;
app.KHzKnob_5.Value = 0;

    apply_filters(app.originalAudio, app.originalSampleRate,
app.frequencyBands, app.filterClass);
    app.ProcessingLabel.Visible = 'off';
end

% Button pushed function: ANALYZEButton
function ANALYZEButtonPushed(app, event)
    app.ProcessingLabel.Visible = 'on';
    print_audio_analysis(app.originalAudio, app.originalSampleRate,
app.frequencyBands, app.filterClass);
    app.ProcessingLabel.Visible = 'off';
end

```

```

        % Close request function: UIFigure
        function UIFigureCloseRequest(app, event)
            delete(app)
            clear sound;
        end
    end

    % Component initialization
    methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 713 747];
app.UIFigure.Name = 'UI Figure';
app.UIFigure.Resize = 'off';
app.UIFigure.CloseRequestFcn = createCallbackFcn(app, @UIFigureCloseRequest,
true);

% Create originalPlot
app.originalPlot = uiaxes(app.UIFigure);
title(app.originalPlot, 'ORIGINAL')
app.originalPlot.XGrid = 'on';
app.originalPlot.YGrid = 'on';
app.originalPlot.Box = 'on';
app.originalPlot.Position = [6 398 339 227];

% Create filteredPlot
app.filteredPlot = uiaxes(app.UIFigure);
title(app.filteredPlot, 'ENHANCED')
app.filteredPlot.XGrid = 'on';
app.filteredPlot.YGrid = 'on';
app.filteredPlot.Box = 'on';
app.filteredPlot.Position = [355 398 355 227];

% Create AUDIOEQUALIZERLabel
app.AUDIOEQUALIZERLabel = uilabel(app.UIFigure);
app.AUDIOEQUALIZERLabel.BackgroundColor = [1 1 1];
app.AUDIOEQUALIZERLabel.HorizontalAlignment = 'center';
app.AUDIOEQUALIZERLabel.FontName = 'SF Mono';
app.AUDIOEQUALIZERLabel.FontSize = 16;
app.AUDIOEQUALIZERLabel.FontWeight = 'bold';
app.AUDIOEQUALIZERLabel.Position = [1 714 713 34];
app.AUDIOEQUALIZERLabel.Text = 'AUDIO EQUALIZER';

```



```

% Create FILTERTYPEButtonGroup
app.FILTERTYPEButtonGroup = uibuttongroup(app.UIFigure);
app.FILTERTYPEButtonGroup.SelectionChangedFcn = createCallbackFcn(app,
@FILTERTYPEButtonGroupSelectionChanged, true);
app.FILTERTYPEButtonGroup.Title = 'FILTER TYPE';
app.FILTERTYPEButtonGroup.BackgroundColor = [1 1 1];
app.FILTERTYPEButtonGroup.FontName = 'MS Sans Serif';
app.FILTERTYPEButtonGroup.Position = [585 629 123 80];

% Create FIRButton
app.FIRButton = uiradiobutton(app.FILTERTYPEButtonGroup);
app.FIRButton.Text = 'FIR';
app.FIRButton.Position = [11 34 58 15];
app.FIRButton.Value = true;

% Create IIRButton
app.IIRButton = uiradiobutton(app.FILTERTYPEButtonGroup);
app.IIRButton.Text = 'IIR';
app.IIRButton.Position = [11 12 65 15];

% Create NOAUDIOFILECHOSENLabel
app.NOAUDIOFILECHOSENLabel = uilabel(app.UIFigure);
app.NOAUDIOFILECHOSENLabel.BackgroundColor = [1 1 1];
app.NOAUDIOFILECHOSENLabel.FontName = 'SF Mono';
app.NOAUDIOFILECHOSENLabel.Position = [32 677 358 22];
app.NOAUDIOFILECHOSENLabel.Text = ' NO AUDIO FILE CHOSEN';

% Create SelectButton
app.SelectButton = uibutton(app.UIFigure, 'push');
app.SelectButton.ButtonPushedFcn = createCallbackFcn(app, @SelectButtonPushed,
true);
app.SelectButton.BackgroundColor = [1 1 1];
app.SelectButton.Position = [401 677 100 22];
app.SelectButton.Text = 'Select';

% Create sampleRateLabel
app.sampleRateLabel = uilabel(app.UIFigure);
app.sampleRateLabel.BackgroundColor = [1 1 1];
app.sampleRateLabel.FontName = 'SF Mono';
app.sampleRateLabel.Position = [32 650 358 22];
app.sampleRateLabel.Text = ' ORIGINAL SAMPLE RATE: 0';

% Create sampleRateField
app.sampleRateField = uieditfield(app.UIFigure, 'numeric');
app.sampleRateField.ValueDisplayFormat = '%.0f';
app.sampleRateField.ValueChangedFcn = createCallbackFcn(app,
@sampleRateFieldValueChanged, true);
app.sampleRateField.HorizontalAlignment = 'center';

```

```

app.sampleRateField.Position = [401 649 100 22];

% Create HzKnobLabel
app.HzKnobLabel = uilabel(app.UIFigure);
app.HzKnobLabel.BackgroundColor = [1 1 1];
app.HzKnobLabel.HorizontalAlignment = 'center';
app.HzKnobLabel.VerticalAlignment = 'top';
app.HzKnobLabel.Position = [112 197 61 15];
app.HzKnobLabel.Text = '0 - 170 Hz';

% Create HzKnob_1
app.HzKnob_1 = uiknob(app.UIFigure, 'continuous');
app.HzKnob_1.Limits = [-24 24];
app.HzKnob_1.ValueChangedFcn = createCallbackFcn(app, @HzKnob_1ValueChanged,
true);
app.HzKnob_1.Position = [112 235 60 60];

% Create HzKnob_10Label
app.HzKnob_10Label = uilabel(app.UIFigure);
app.HzKnob_10Label.BackgroundColor = [1 1 1];
app.HzKnob_10Label.HorizontalAlignment = 'center';
app.HzKnob_10Label.VerticalAlignment = 'top';
app.HzKnob_10Label.Position = [252 197 85 15];
app.HzKnob_10Label.Text = '170 - 310 Hz';

% Create HzKnob_2
app.HzKnob_2 = uiknob(app.UIFigure, 'continuous');
app.HzKnob_2.Limits = [-24 24];
app.HzKnob_2.ValueChangedFcn = createCallbackFcn(app, @HzKnob_2ValueChanged,
true);
app.HzKnob_2.Position = [263 235 60 60];

% Create HzLabel
app.HzLabel = uilabel(app.UIFigure);
app.HzLabel.BackgroundColor = [1 1 1];
app.HzLabel.HorizontalAlignment = 'center';
app.HzLabel.VerticalAlignment = 'top';
app.HzLabel.Position = [407 196 75 15];
app.HzLabel.Text = '310 - 600 Hz';

% Create HzKnob_3
app.HzKnob_3 = uiknob(app.UIFigure, 'continuous');
app.HzKnob_3.Limits = [-24 24];
app.HzKnob_3.ValueChangedFcn = createCallbackFcn(app, @HzKnob_3ValueChanged,
true);
app.HzKnob_3.Position = [414 235 60 60];

```

```

% Create HzKnob_11Label
app.HzKnob_11Label = uilabel(app.UIFigure);
app.HzKnob_11Label.BackgroundColor = [1 1 1];
app.HzKnob_11Label.HorizontalAlignment = 'center';
app.HzKnob_11Label.VerticalAlignment = 'top';
app.HzKnob_11Label.Position = [545 196 81 15];
app.HzKnob_11Label.Text = '600 - 1000 Hz';

% Create HzKnob_4
app.HzKnob_4 = uiknob(app.UIFigure, 'continuous');
app.HzKnob_4.Limits = [-24 24];
app.HzKnob_4.ValueChangedFcn = createCallbackFcn(app, @HzKnob_4ValueChanged,
true);
app.HzKnob_4.Position = [555 236 60 60];

% Create KHzKnobLabel
app.KHzKnobLabel = uilabel(app.UIFigure);
app.KHzKnobLabel.BackgroundColor = [1 1 1];
app.KHzKnobLabel.HorizontalAlignment = 'center';
app.KHzKnobLabel.VerticalAlignment = 'top';
app.KHzKnobLabel.Position = [55 55 56 15];
app.KHzKnobLabel.Text = '1 - 3 KHz';

% Create KHzKnob_1
app.KHzKnob_1 = uiknob(app.UIFigure, 'continuous');
app.KHzKnob_1.Limits = [-24 24];
app.KHzKnob_1.ValueChangedFcn = createCallbackFcn(app, @KHzKnob_1ValueChanged,
true);
app.KHzKnob_1.Position = [52 95 60 60];

% Create KHzKnob_2Label
app.KHzKnob_2Label = uilabel(app.UIFigure);
app.KHzKnob_2Label.BackgroundColor = [1 1 1];
app.KHzKnob_2Label.HorizontalAlignment = 'center';
app.KHzKnob_2Label.VerticalAlignment = 'top';
app.KHzKnob_2Label.Position = [197 54 56 15];
app.KHzKnob_2Label.Text = '3 - 6 KHz';

% Create KHzKnob_2
app.KHzKnob_2 = uiknob(app.UIFigure, 'continuous');
app.KHzKnob_2.Limits = [-24 24];
app.KHzKnob_2.ValueChangedFcn = createCallbackFcn(app, @KHzKnob_2ValueChanged,
true);
app.KHzKnob_2.Position = [194 95 60 60];

% Create KHzKnob_3Label
app.KHzKnob_3Label = uilabel(app.UIFigure);
app.KHzKnob_3Label.BackgroundColor = [1 1 1];

```

```

app.KHzKnob_3Label.HorizontalAlignment = 'center';
app.KHzKnob_3Label.VerticalAlignment = 'top';
app.KHzKnob_3Label.Position = [336 54 63 15];
app.KHzKnob_3Label.Text = '6 - 12 KHz';

% Create KHzKnob_3
app.KHzKnob_3 = uiknob(app.UIFigure, 'continuous');
app.KHzKnob_3.Limits = [-24 24];
app.KHzKnob_3.ValueChangedFcn = createCallbackFcn(app, @KHzKnob_3ValueChanged,
true);
app.KHzKnob_3.Position = [337 93 60 60];

% Create KHzKnob_4Label
app.KHzKnob_4Label = uilabel(app.UIFigure);
app.KHzKnob_4Label.BackgroundColor = [1 1 1];
app.KHzKnob_4Label.HorizontalAlignment = 'center';
app.KHzKnob_4Label.VerticalAlignment = 'top';
app.KHzKnob_4Label.Position = [480 53 69 15];
app.KHzKnob_4Label.Text = '12 - 14 KHz';

% Create KHzKnob_4
app.KHzKnob_4 = uiknob(app.UIFigure, 'continuous');
app.KHzKnob_4.Limits = [-24 24];
app.KHzKnob_4.ValueChangedFcn = createCallbackFcn(app, @KHzKnob_4ValueChanged,
true);
app.KHzKnob_4.Position = [484 93 60 60];

% Create KHzKnob_5Label
app.KHzKnob_5Label = uilabel(app.UIFigure);
app.KHzKnob_5Label.BackgroundColor = [1 1 1];
app.KHzKnob_5Label.HorizontalAlignment = 'center';
app.KHzKnob_5Label.VerticalAlignment = 'top';
app.KHzKnob_5Label.Position = [615 53 69 15];
app.KHzKnob_5Label.Text = '14 - 16 KHz';

% Create KHzKnob_5
app.KHzKnob_5 = uiknob(app.UIFigure, 'continuous');
app.KHzKnob_5.Limits = [-24 24];
app.KHzKnob_5.ValueChangedFcn = createCallbackFcn(app, @KHzKnob_5ValueChanged,
true);
app.KHzKnob_5.Position = [619 95 60 60];

% Create APPLYFILTERSButton
app.APPLYFILTERSButton = uibutton(app.UIFigure, 'push');
app.APPLYFILTERSButton.ButtonPushedFcn = createCallbackFcn(app,
@APPLYFILTERSButtonPushed, true);
app.APPLYFILTERSButton.Position = [29 349 101 22];
app.APPLYFILTERSButton.Text = 'APPLY FILTERS';

```

```

% Create RESETButton
app.RESETButton = uibutton(app.UIFigure, 'push');
app.RESETButton.ButtonPushedFcn = createCallbackFcn(app, @RESETButtonPushed,
true);
app.RESETButton.Position = [149 349 104 22];
app.RESETButton.Text = 'RESET';

% Create playButton
app.playButton = uibutton(app.UIFigure, 'push');
app.playButton.ButtonPushedFcn = createCallbackFcn(app, @playButtonPushed,
true);
app.playButton.Icon = 'play.png';
app.playButton.IconAlignment = 'center';
app.playButton.Position = [283 332 72 54];
app.playButton.Text = '';

% Create stopButton
app.stopButton = uibutton(app.UIFigure, 'push');
app.stopButton.ButtonPushedFcn = createCallbackFcn(app, @stopButtonPushed,
true);
app.stopButton.Icon = 'stop.png';
app.stopButton.IconAlignment = 'center';
app.stopButton.Position = [364 332 72 54];
app.stopButton.Text = '';

% Create SAVEButton
app.SAVEButton = uibutton(app.UIFigure, 'push');
app.SAVEButton.ButtonPushedFcn = createCallbackFcn(app, @SAVEButtonPushed,
true);
app.SAVEButton.Position = [592 349 100 22];
app.SAVEButton.Text = 'SAVE';

% Create FrequencyBands
app.FrequencyBands = uilabel(app.UIFigure);
app.FrequencyBands.HorizontalAlignment = 'center';
app.FrequencyBands.FontName = 'Monaco';
app.FrequencyBands.Enable = 'off';
app.FrequencyBands.Visible = 'off';
app.FrequencyBands.Position = [204 10 309 15];
app.FrequencyBands.Text = '[1 1 1 1 1 1 1 1 1]';

% Create HzKnob_1_Detail
app.HzKnob_1_Detail = uilabel(app.UIFigure);
app.HzKnob_1_Detail.BackgroundColor = [1 1 1];
app.HzKnob_1_Detail.HorizontalAlignment = 'center';
app.HzKnob_1_Detail.VerticalAlignment = 'top';
app.HzKnob_1_Detail.Position = [112 179 61 15];

```

```

app.HzKnob_1_Detail.Text = '1';

% Create HzKnob_2_Detail
app.HzKnob_2_Detail = uilabel(app.UIFigure);
app.HzKnob_2_Detail.BackgroundColor = [1 1 1];
app.HzKnob_2_Detail.HorizontalAlignment = 'center';
app.HzKnob_2_Detail.VerticalAlignment = 'top';
app.HzKnob_2_Detail.Position = [252 179 85 15];
app.HzKnob_2_Detail.Text = '1';

% Create HzKnob_3_Detail
app.HzKnob_3_Detail = uilabel(app.UIFigure);
app.HzKnob_3_Detail.BackgroundColor = [1 1 1];
app.HzKnob_3_Detail.HorizontalAlignment = 'center';
app.HzKnob_3_Detail.VerticalAlignment = 'top';
app.HzKnob_3_Detail.Position = [407 179 74 15];
app.HzKnob_3_Detail.Text = '1';

% Create HzKnob_4_Detail
app.HzKnob_4_Detail = uilabel(app.UIFigure);
app.HzKnob_4_Detail.BackgroundColor = [1 1 1];
app.HzKnob_4_Detail.HorizontalAlignment = 'center';
app.HzKnob_4_Detail.VerticalAlignment = 'top';
app.HzKnob_4_Detail.Position = [548 178 78 15];
app.HzKnob_4_Detail.Text = '1';

% Create KHzKnob_1_Detail
app.KHzKnob_1_Detail = uilabel(app.UIFigure);
app.KHzKnob_1_Detail.BackgroundColor = [1 1 1];
app.KHzKnob_1_Detail.HorizontalAlignment = 'center';
app.KHzKnob_1_Detail.VerticalAlignment = 'top';
app.KHzKnob_1_Detail.Position = [55 37 56 15];
app.KHzKnob_1_Detail.Text = '1';

% Create KHzKnob_2_Detail
app.KHzKnob_2_Detail = uilabel(app.UIFigure);
app.KHzKnob_2_Detail.BackgroundColor = [1 1 1];
app.KHzKnob_2_Detail.HorizontalAlignment = 'center';
app.KHzKnob_2_Detail.VerticalAlignment = 'top';
app.KHzKnob_2_Detail.Position = [197 37 56 15];
app.KHzKnob_2_Detail.Text = '1';

% Create KHzKnob_3_Detail
app.KHzKnob_3_Detail = uilabel(app.UIFigure);
app.KHzKnob_3_Detail.BackgroundColor = [1 1 1];
app.KHzKnob_3_Detail.HorizontalAlignment = 'center';
app.KHzKnob_3_Detail.VerticalAlignment = 'top';
app.KHzKnob_3_Detail.Position = [336 37 63 15];

```

```

app.KHzKnob_3_Detail.Text = '1';

% Create KHzKnob_4_Detail
app.KHzKnob_4_Detail = uilabel(app.UIFigure);
app.KHzKnob_4_Detail.BackgroundColor = [1 1 1];
app.KHzKnob_4_Detail.HorizontalAlignment = 'center';
app.KHzKnob_4_Detail.VerticalAlignment = 'top';
app.KHzKnob_4_Detail.Position = [481 36 68 15];
app.KHzKnob_4_Detail.Text = '1';

% Create KHzKnob_5_Detail
app.KHzKnob_5_Detail = uilabel(app.UIFigure);
app.KHzKnob_5_Detail.BackgroundColor = [1 1 1];
app.KHzKnob_5_Detail.HorizontalAlignment = 'center';
app.KHzKnob_5_Detail.VerticalAlignment = 'top';
app.KHzKnob_5_Detail.Position = [615 36 69 15];
app.KHzKnob_5_Detail.Text = '1';

% Create ANALYZEButton
app.ANALYZEButton = uibutton(app.UIFigure, 'push');
app.ANALYZEButton.ButtonPushedFcn = createCallbackFcn(app,
@ANALYZEButtonPushed, true);
app.ANALYZEButton.Position = [473 348 100 23];
app.ANALYZEButton.Text = 'ANALYZE';

% Create ProcessingLabel
app.ProcessingLabel = uilabel(app.UIFigure);
app.ProcessingLabel.HorizontalAlignment = 'center';
app.ProcessingLabel.FontSize = 16;
app.ProcessingLabel.FontWeight = 'bold';
app.ProcessingLabel.FontColor = [0.149 0.149 0.149];
app.ProcessingLabel.Visible = 'off';
app.ProcessingLabel.Position = [276 612 161 30];
app.ProcessingLabel.Text = 'Processing...';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = AudioEqualizer

```

```

runningApp = getRunningApp(app);

% Check for running singleton app
if isempty(runningApp)

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app UIFigure)

% Execute the startup function
runStartupFcn(app, @startupFcn)
else

% Focus the running singleton app
figure(runningApp UIFigure)

app = runningApp;
end

if nargin == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app UIFigure)
end
end
end

```


apply_filters

```
function resultData = apply_filters(audioData,
sampleRate, gains, filterClass)

firstFreq = [0 170 310 600 1000 3000 6000 12000 14000];
secondFreq = [170 310 600 1000 3000 6000 12000 14000
16000];
order = 0;

if filterClass == 0 % FIR
    order = 100;
elseif filterClass == 1 % IIR
    order = 4;
end

[b, a] = low_pass(170, sampleRate, filterClass, order);
y = filter(b, a, audioData);
resultData = y * gains(1);

for i=2:9
    [b, a] = band_pass(firstFreq(i), secondFreq(i),
sampleRate, filterClass, order);
    y = filter(b, a, audioData);
    resultData = resultData + y * gains(i);
end
end
```

low_pass

```
function [b, a] = low_pass(Fc, sampleRate, filterClass,
order)
% This function returns the numerator and denominator of
the low-pass filter
% @filterClass = 0 : FIR , @filterClass = 1 : IIR

Fm = sampleRate/2; % Nyquist Freq

if(filterClass == 0) % FIR
    [b, a] = fir1(order, Fc/Fm, 'low');
    [b, a] = eqtflength(b, a);
elseif(filterClass == 1) % IIR
    [b, a] = butter(order, Fc/Fm, 'low');
    [b, a] = eqtflength(b, a);
end
end
```

band_pass

```
function [b, a] = band_pass(Fc1, Fc2, sampleRate,
filterClass, order)
% This function returns the numerator and denominator of
the bandpass filter
% @filterClass = 0 : FIR , @filterClass = 1 : IIR

Fm = sampleRate/2; % Nyquist freq

if(filterClass == 0) % FIR
    [b, a] = fir1(order, [Fc1 Fc2]/Fm, 'bandpass');
    [b, a] = eqtflength(b, a);
elseif(filterClass == 1) % IIR
    [b, a] = butter(order, [Fc1 Fc2]/Fm, 'bandpass');
    [b, a] = eqtflength(b,a);
end

end
```

print_audio_analysis

```
function figures = print_audio_analysis(audioData,
sampleRate, gains, filterClass)
%This function generates subplots relating to the audio
wave we are working
%with

firstFreq = [0 170 310 600 1000 3000 6000 12000 14000];
secondFreq = [170 310 600 1000 3000 6000 12000 14000
16000];
order = 0;
samples = length(audioData);
t = linspace(0,samples/sampleRate,samples);
f = linspace(-sampleRate/2,sampleRate/2,samples);
figures = [];

if filterClass == 0 % FIR
    order = 100;
elseif filterClass == 1 % IIR
    order = 4;
end

[b, a] = low_pass(170, sampleRate, filterClass, order);
y = filter(b, a, audioData);
```

```

resultData = y * gains(1);

% Filter analysis
TF = tf(b, a);

fig = figure();
freqz(b, a, samples);
title('Magnitude & Phase of H1(Z) (0 - 170)');
figures = [figures fig];

fig = figure();
subplot(3, 1, 1);
pzmap(TF);
title('Zeros & Poles of H1(Z) (0 - 170)');
figures = [figures fig];

subplot(3, 1, 2);
impz(TF);
title('Impulse response h1(n)');

subplot(3, 1, 3);
step(TF);
title('Step response s1(n)')

%Output analysis
y = filter(b, a, audioData);
Y = fftshift(fft(y/sampleRate));

fig = figure();
subplot(3, 1, 1);
plot(t,y);
title('Time domain of y1(n)');
figures = [figures fig];

subplot(3, 1, 2);
plot(f, abs(Y));
title('Magnitude of Y1(w)');

subplot(3, 1, 3);
plot(f, angle(Y));
title('Phase of Y1(w)');

for i=2:9
    [b, a] = band_pass(firstFreq(i), secondFreq(i),
sampleRate, filterClass, order);
    y = filter(b, a, audioData);
    resultData = resultData + y * gains(i);

```

```

%Filter Analysis
TF = tf(b, a);

fig = figure();
freqz(b, a, samples);
range = strcat(' (',int2str(firstFreq(i)), 'Hz -
',int2str(secondFreq(i)), 'Hz) ');
title(strcat('Magnitude & Phase of
H',int2str(i), '(Z)',range));
figures = [figures fig];

fig = figure();
subplot(3, 1, 1);
pzmap(TF);
title(strcat('Zeros & Poles of
H',int2str(i), '(Z)',range));
figures = [figures fig];

subplot(3, 1, 2);
step(TF);
title(strcat('Step response S',int2str(i), '(n)'));

subplot(3, 1, 3);
impulse(TF);
title(strcat('Impulse response h',int2str(i), '(n)'));

%Output Analysis
y = filter(b, a, audioData);
Y = fftshift(fft(y/sampleRate));

fig = figure();
subplot(3, 1, 1);
plot(t,y);
title(strcat('Time domain y', int2str(i), '(n)',
range));
figures = [figures fig];

subplot(3, 1, 2);
plot(f, abs(Y));
title(strcat('Magnitude of Y', int2str(i), '(w)'));

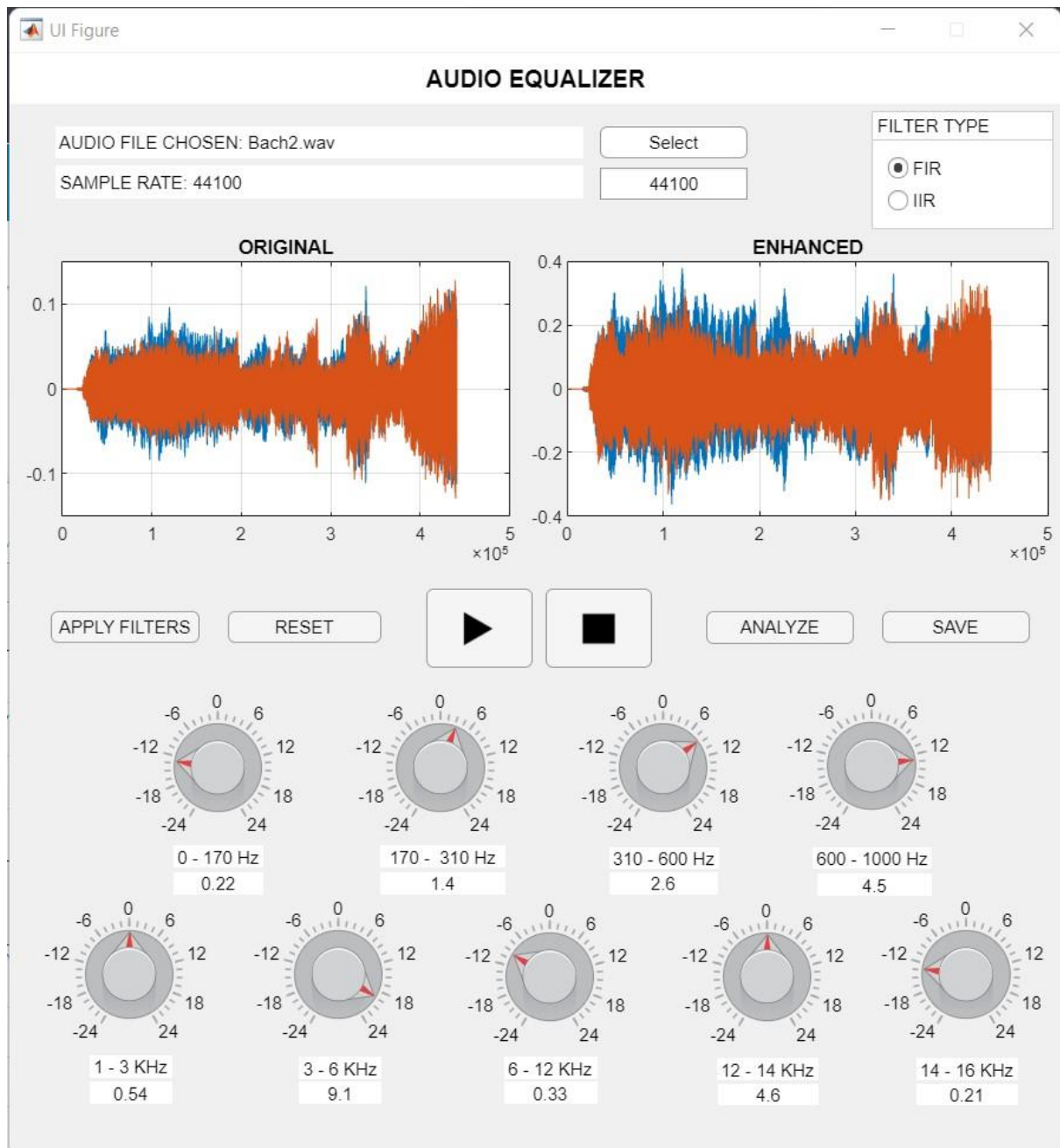
subplot(3, 1, 3);
plot(f, angle(Y));
title(strcat('Phase of Y', int2str(i), '(w)'));
end

end

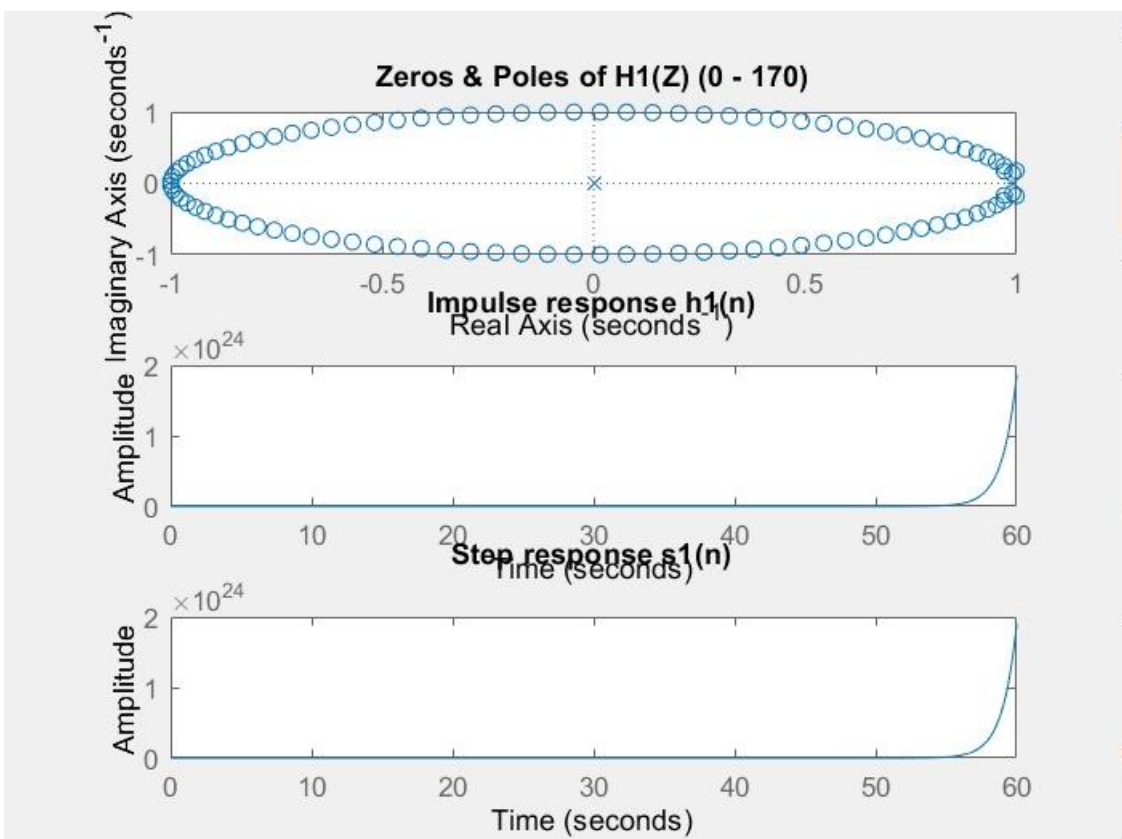
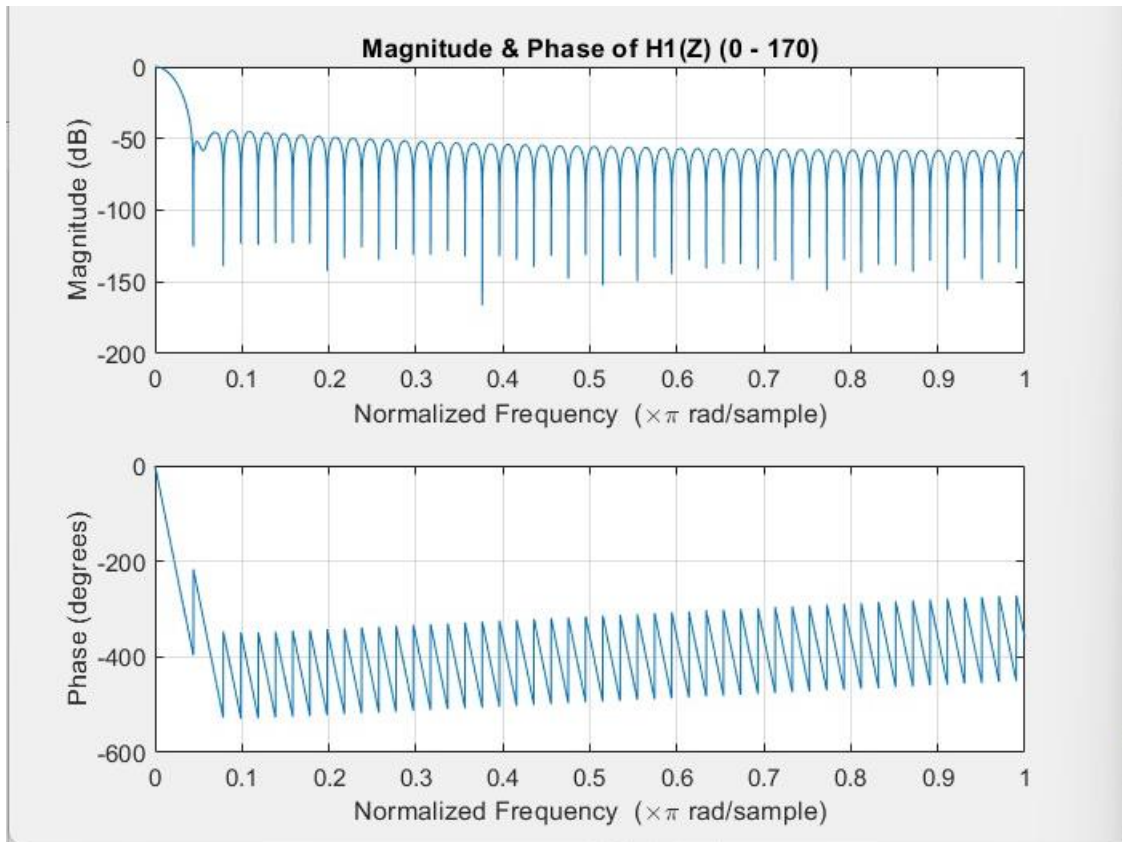
```

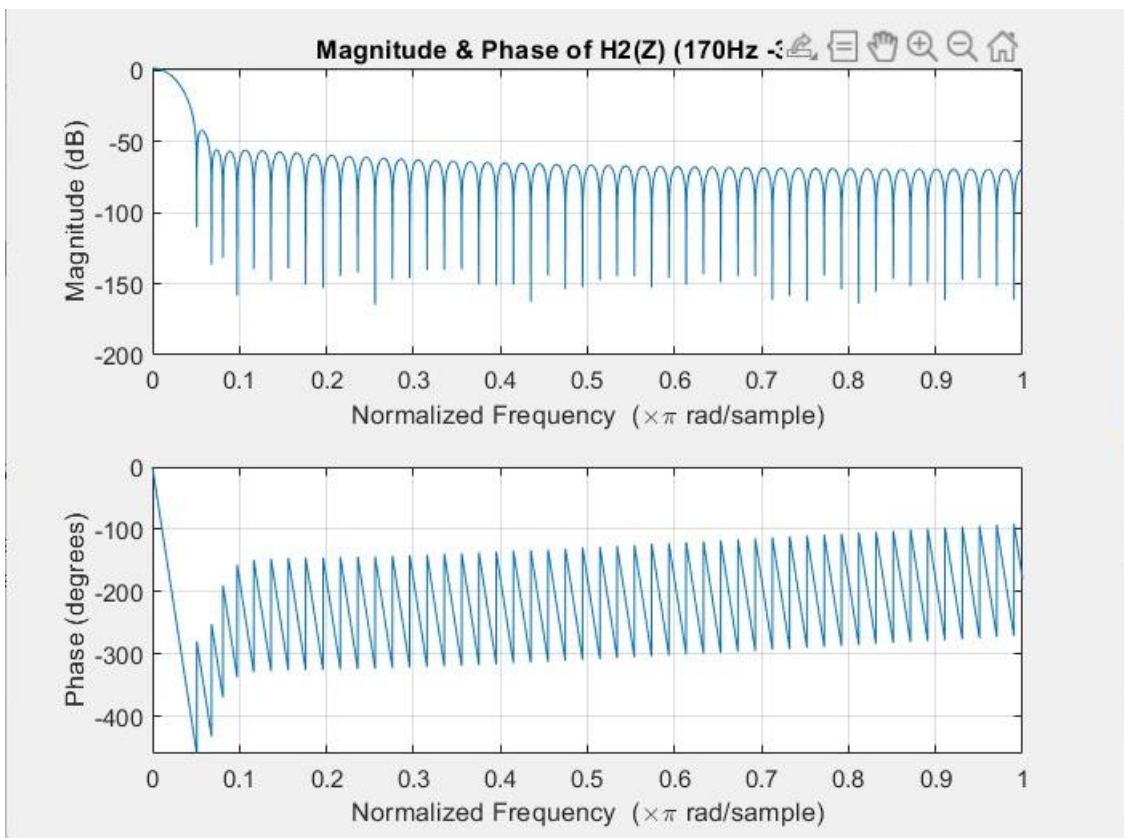
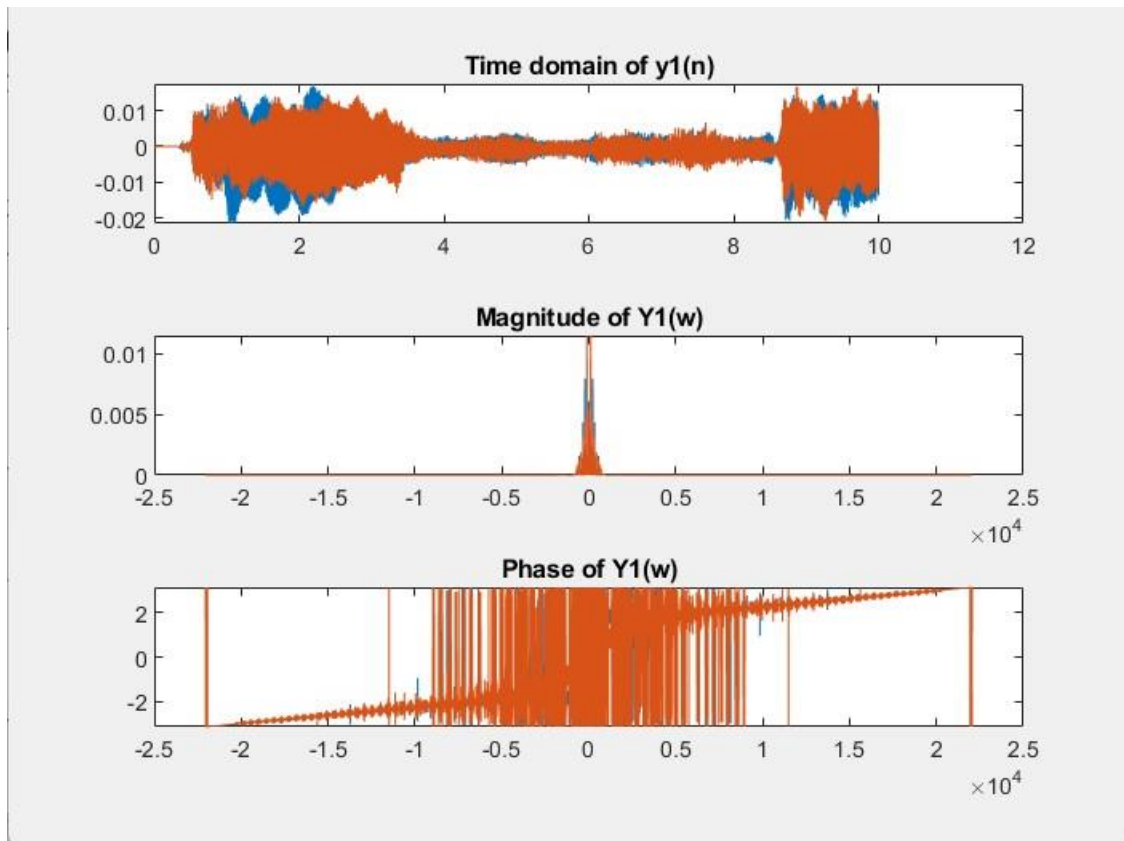
Sample Run

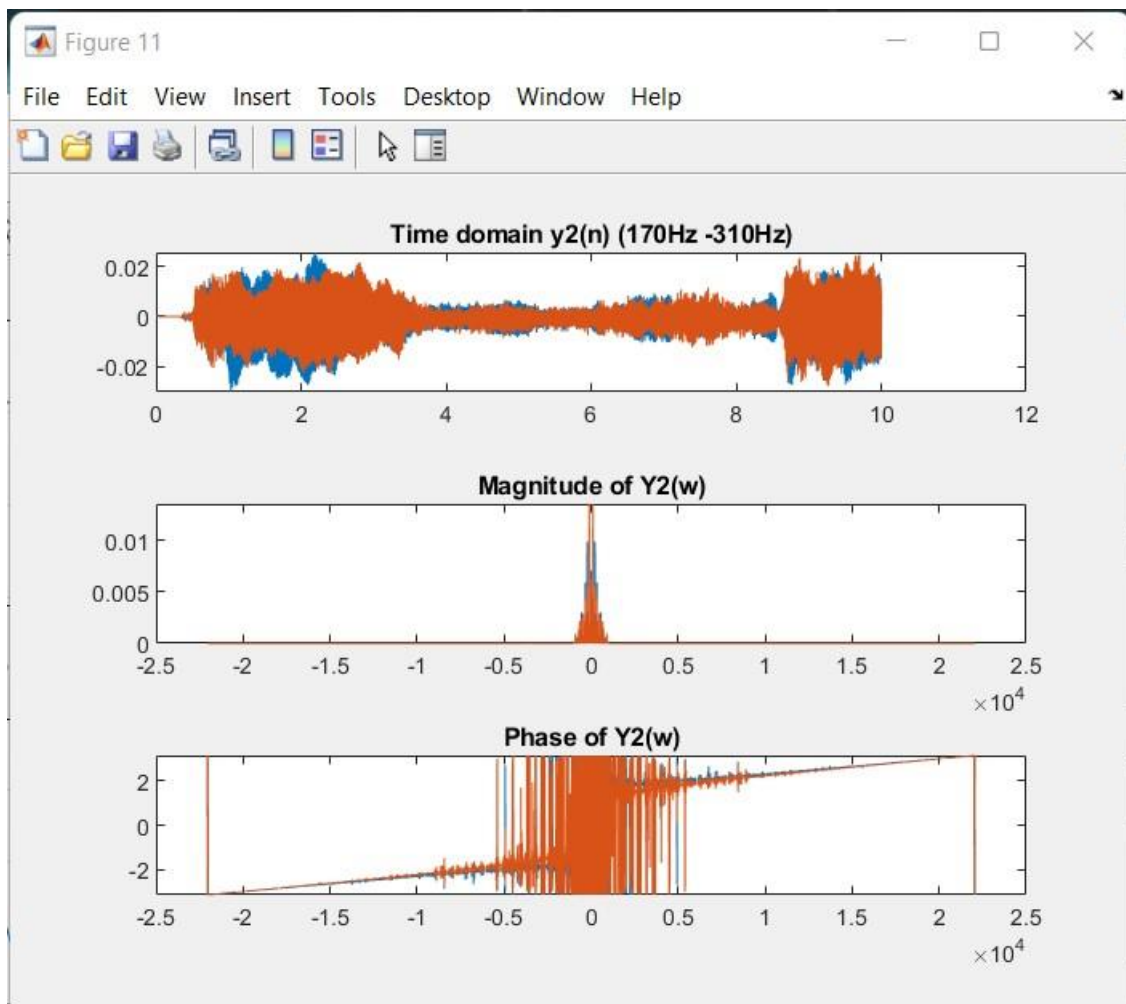
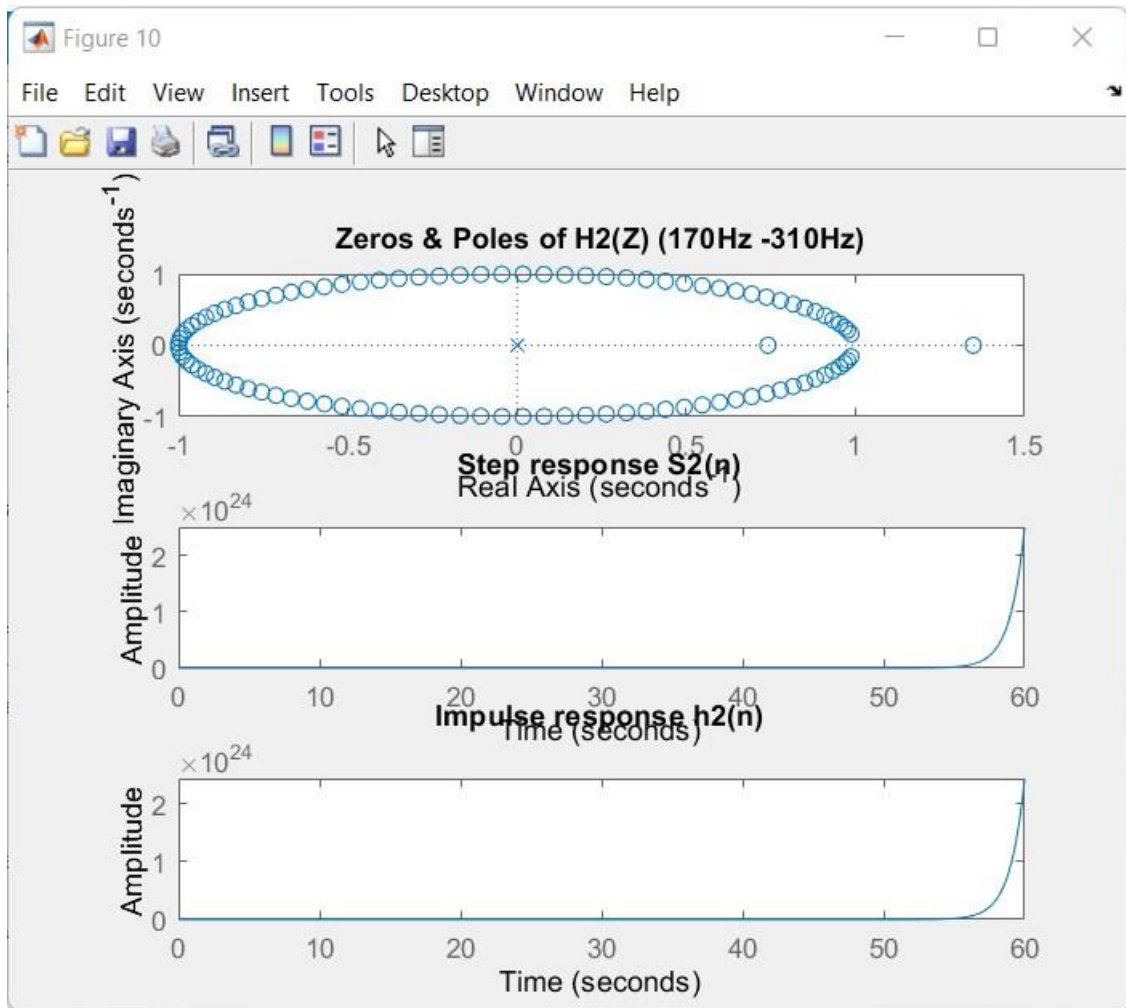
Main User Interface

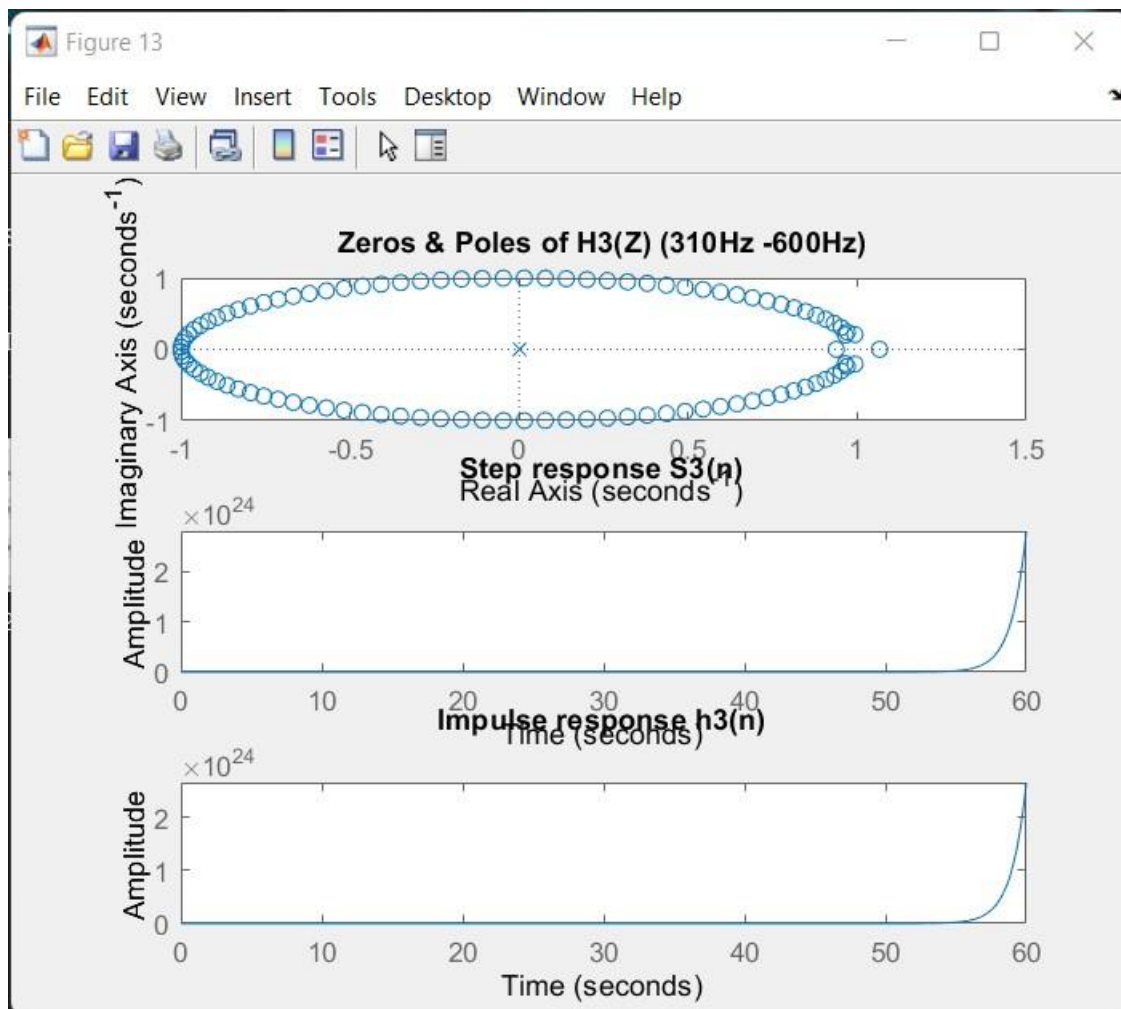
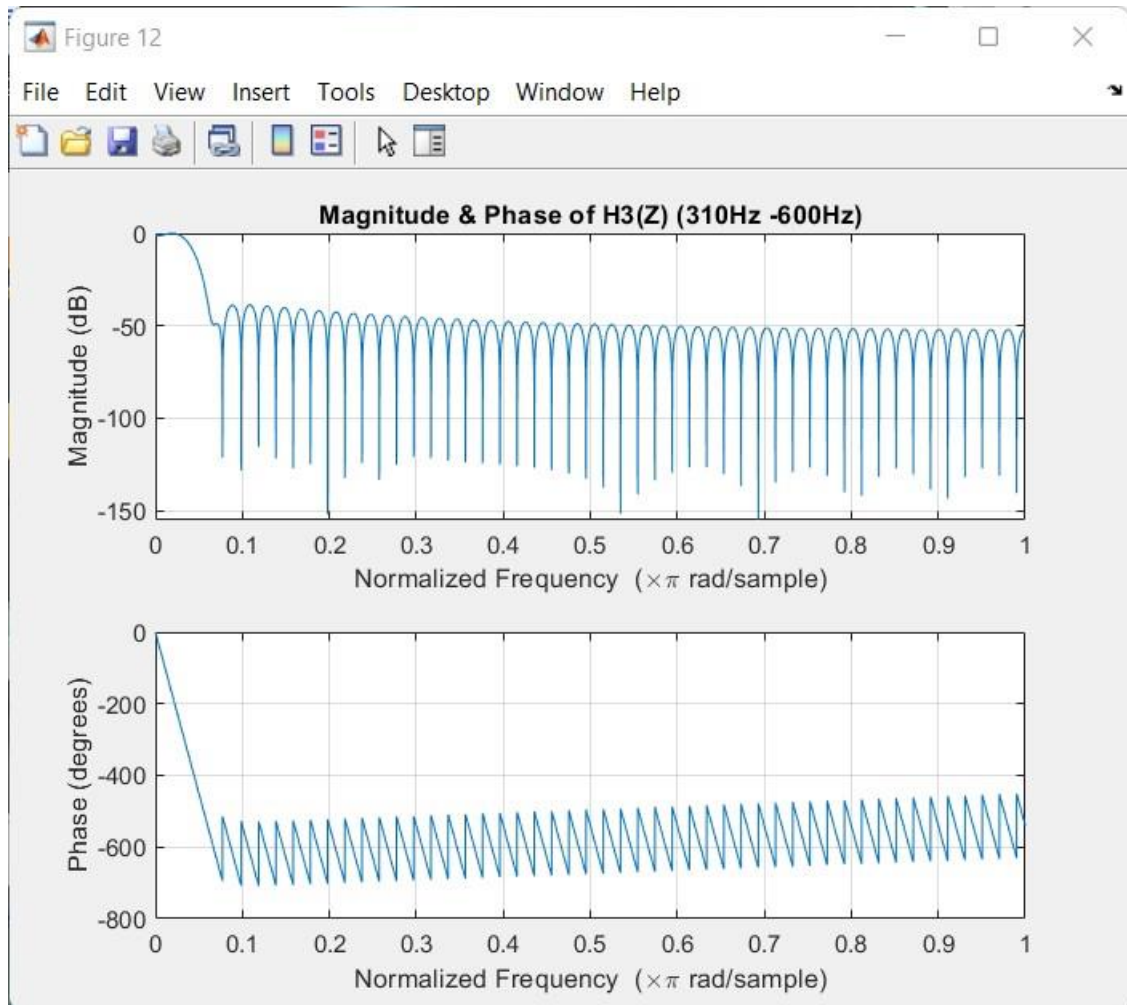


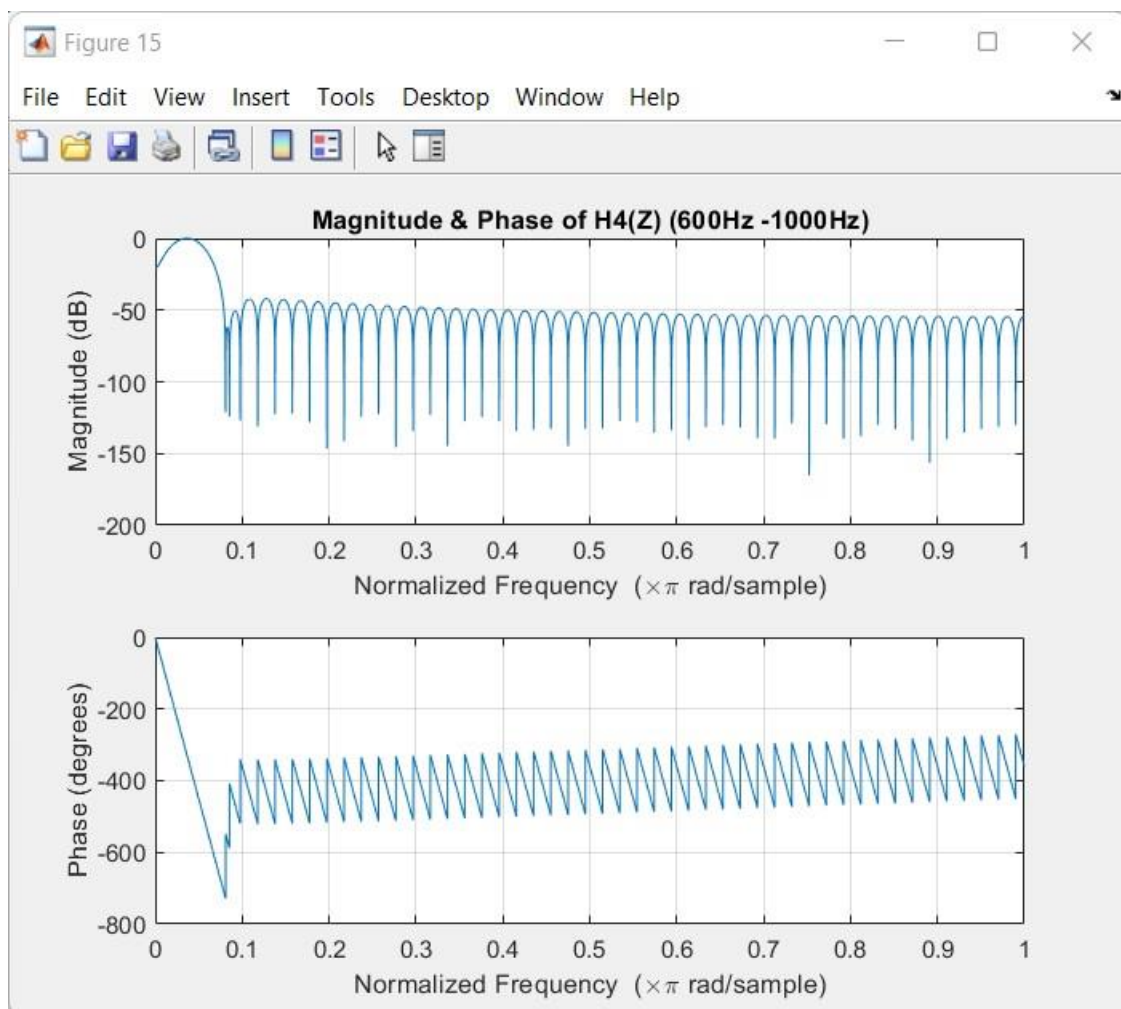
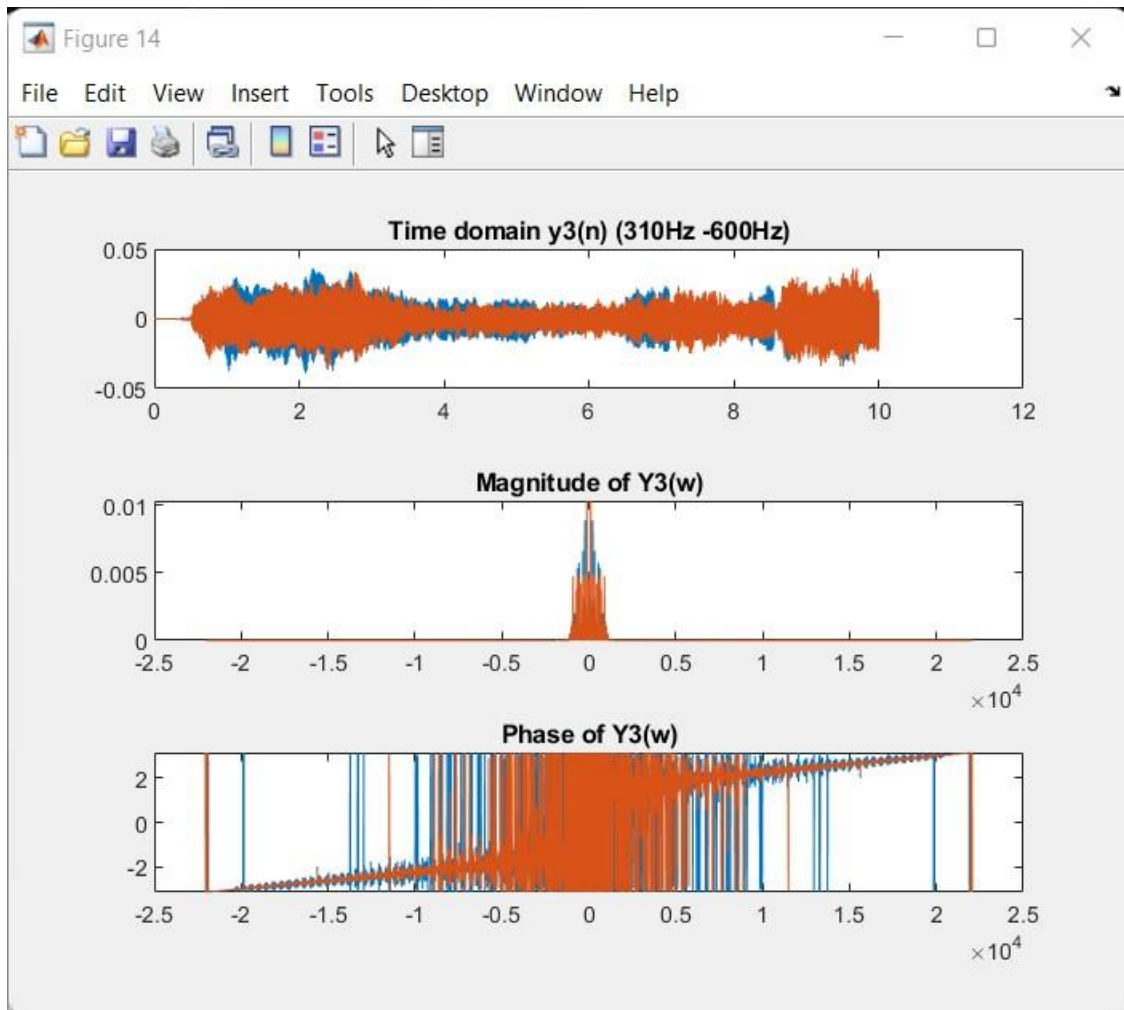
FIR Analysis

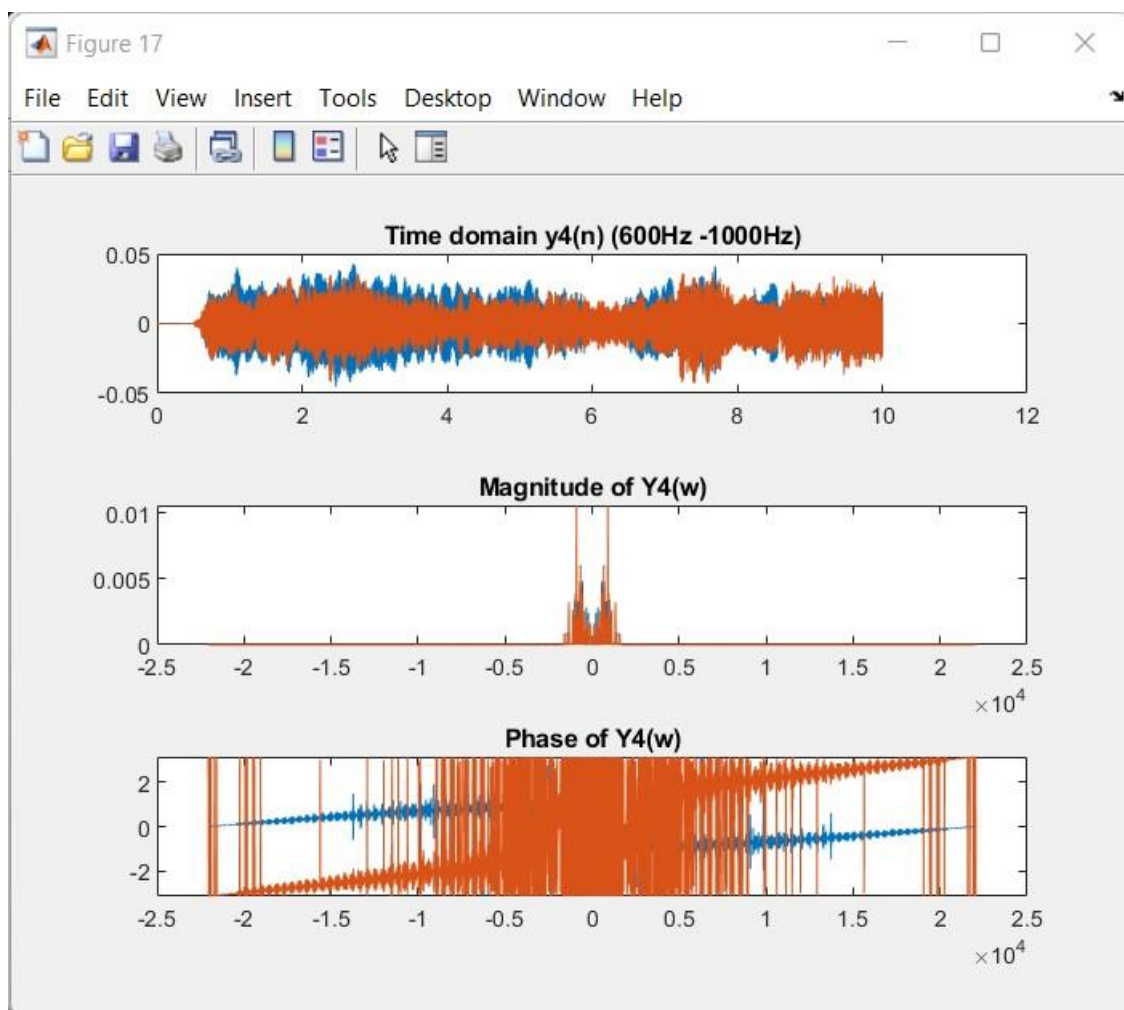
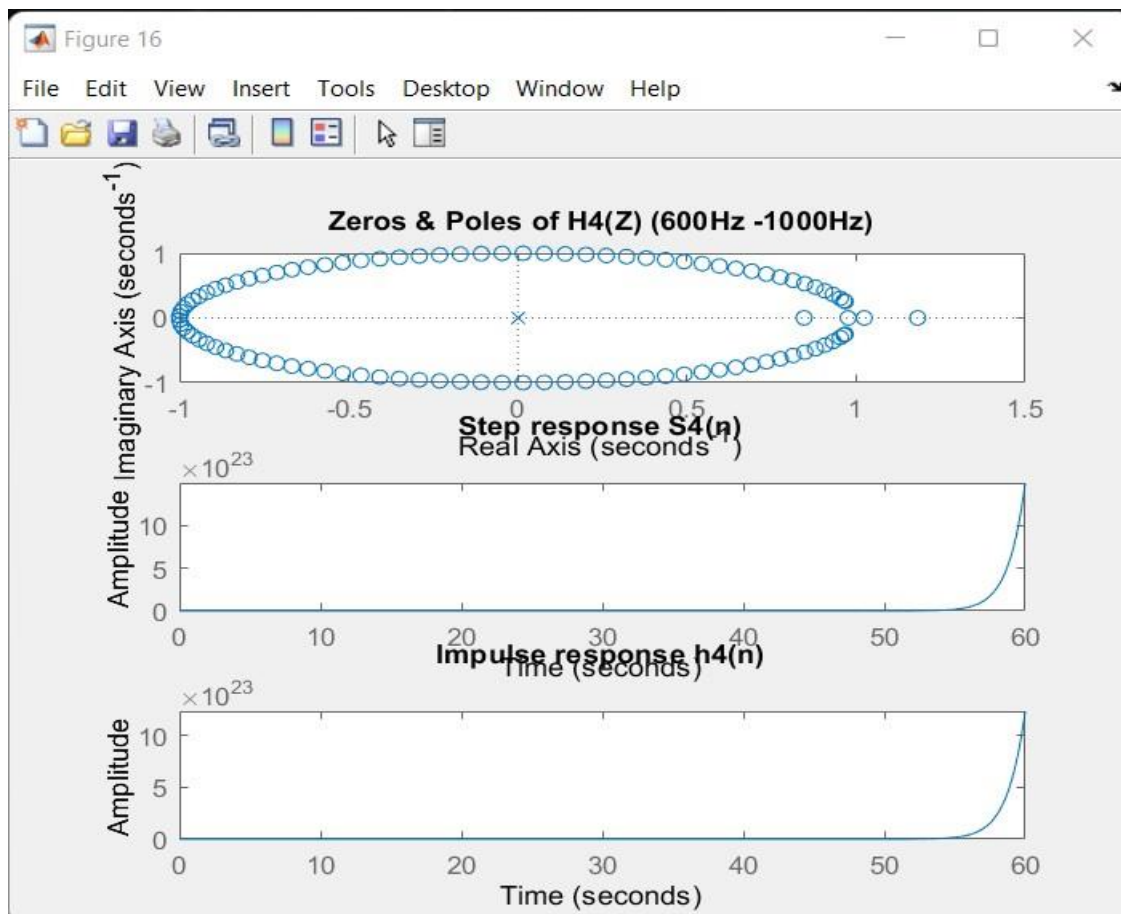


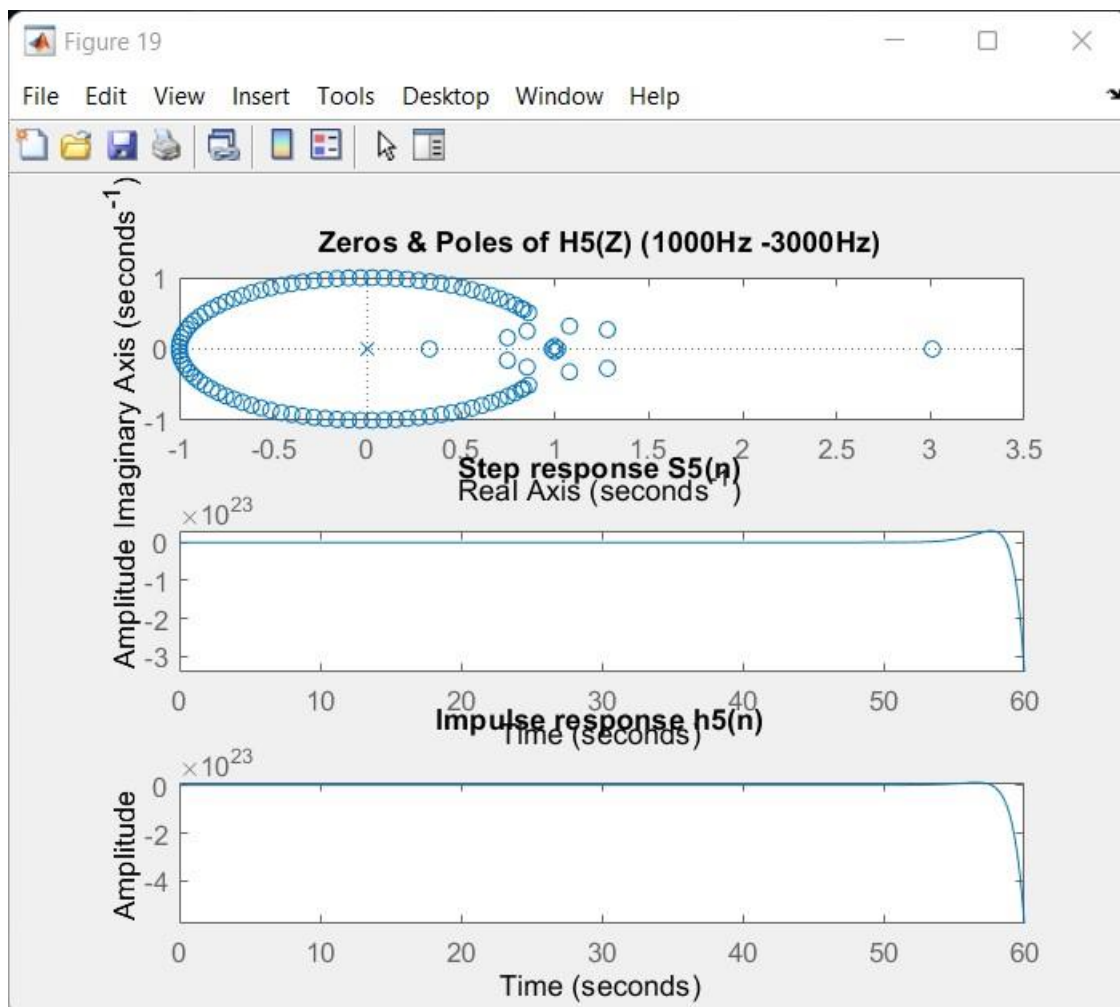
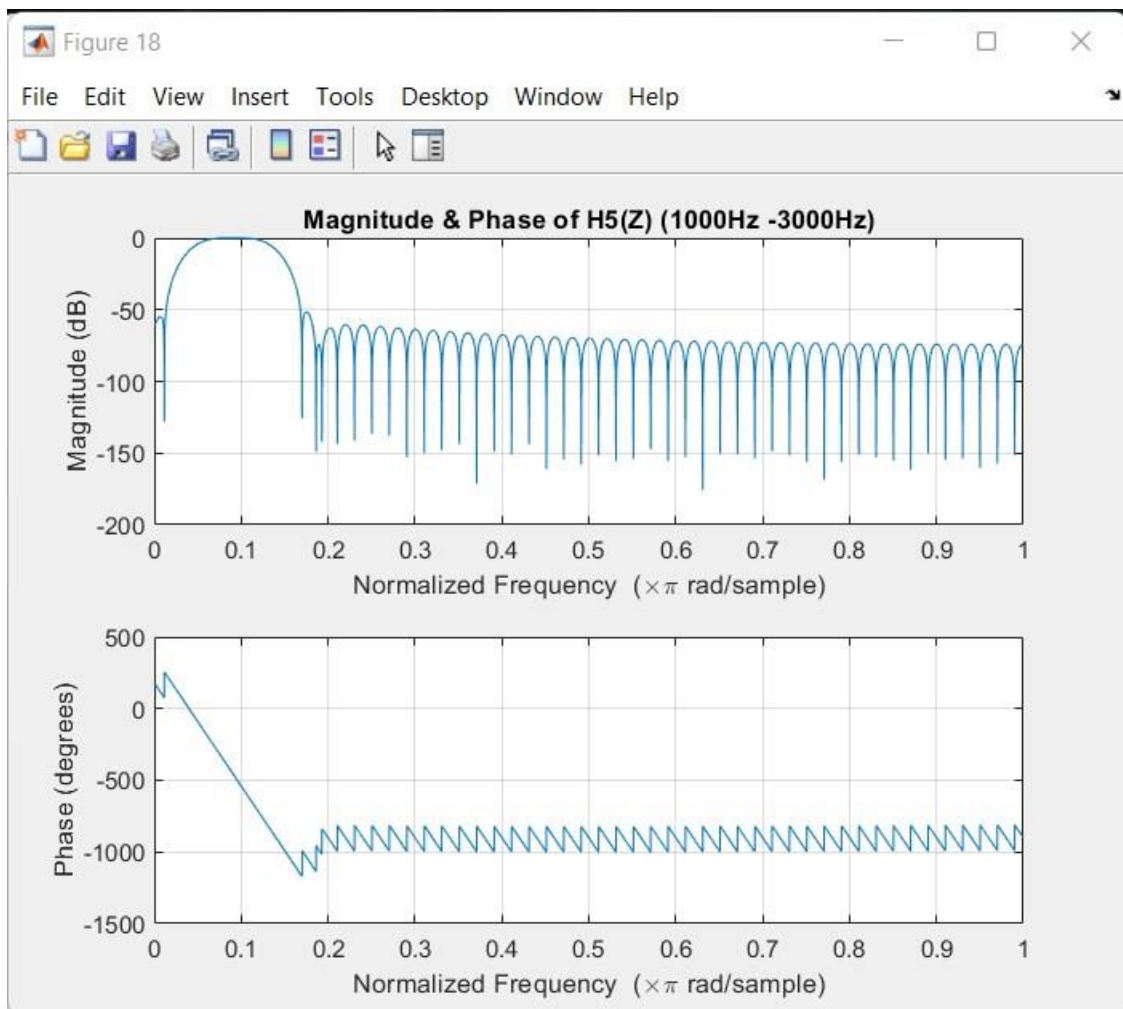


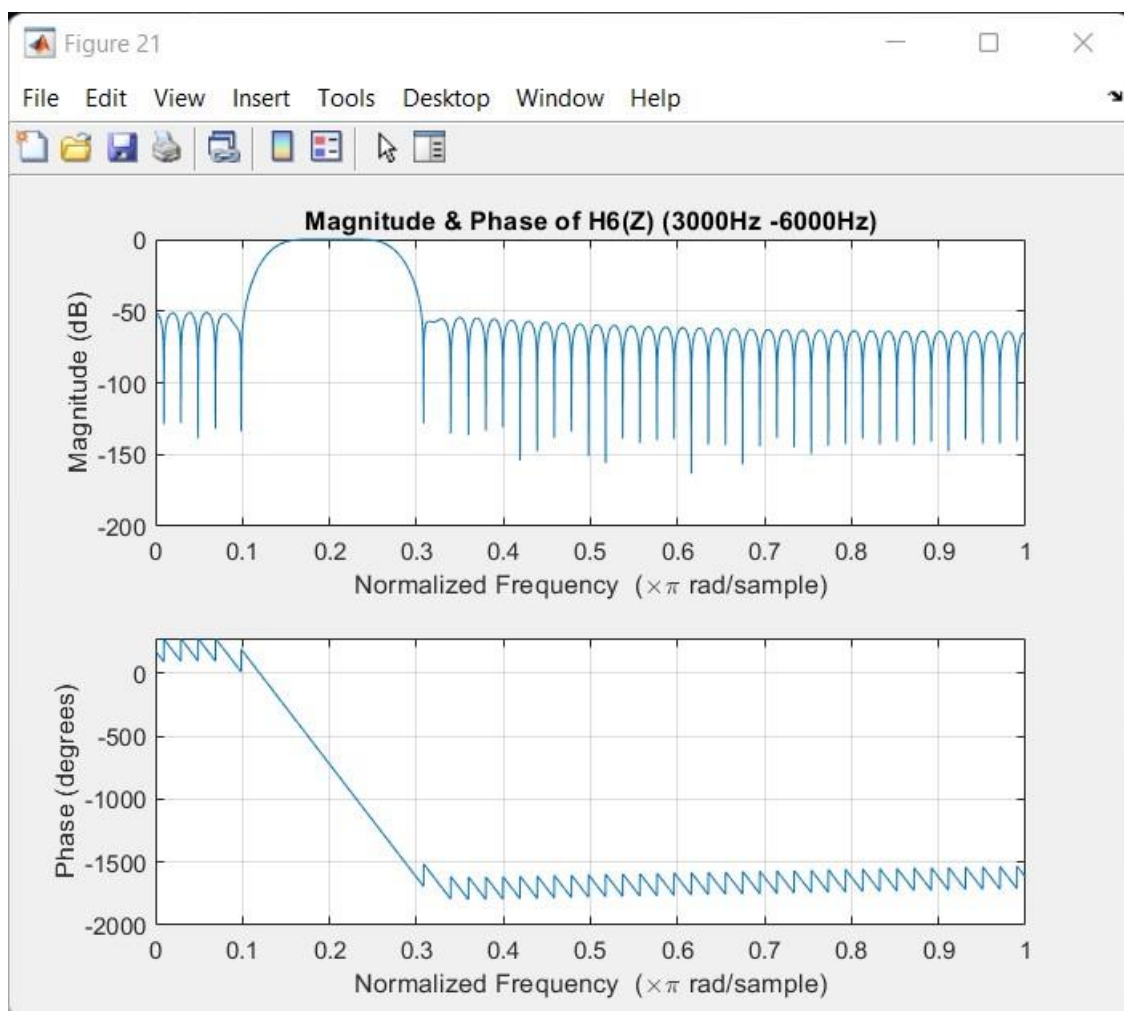
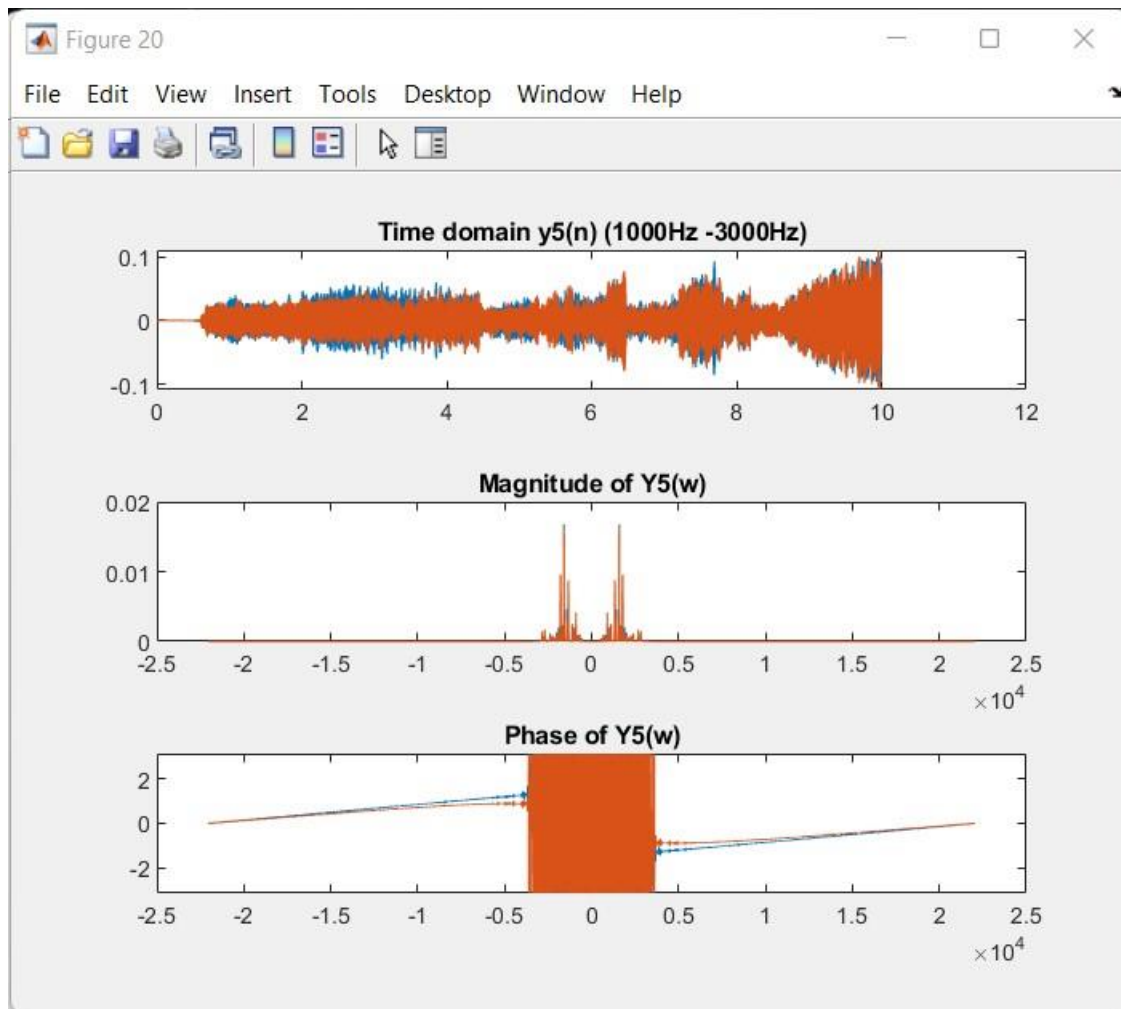


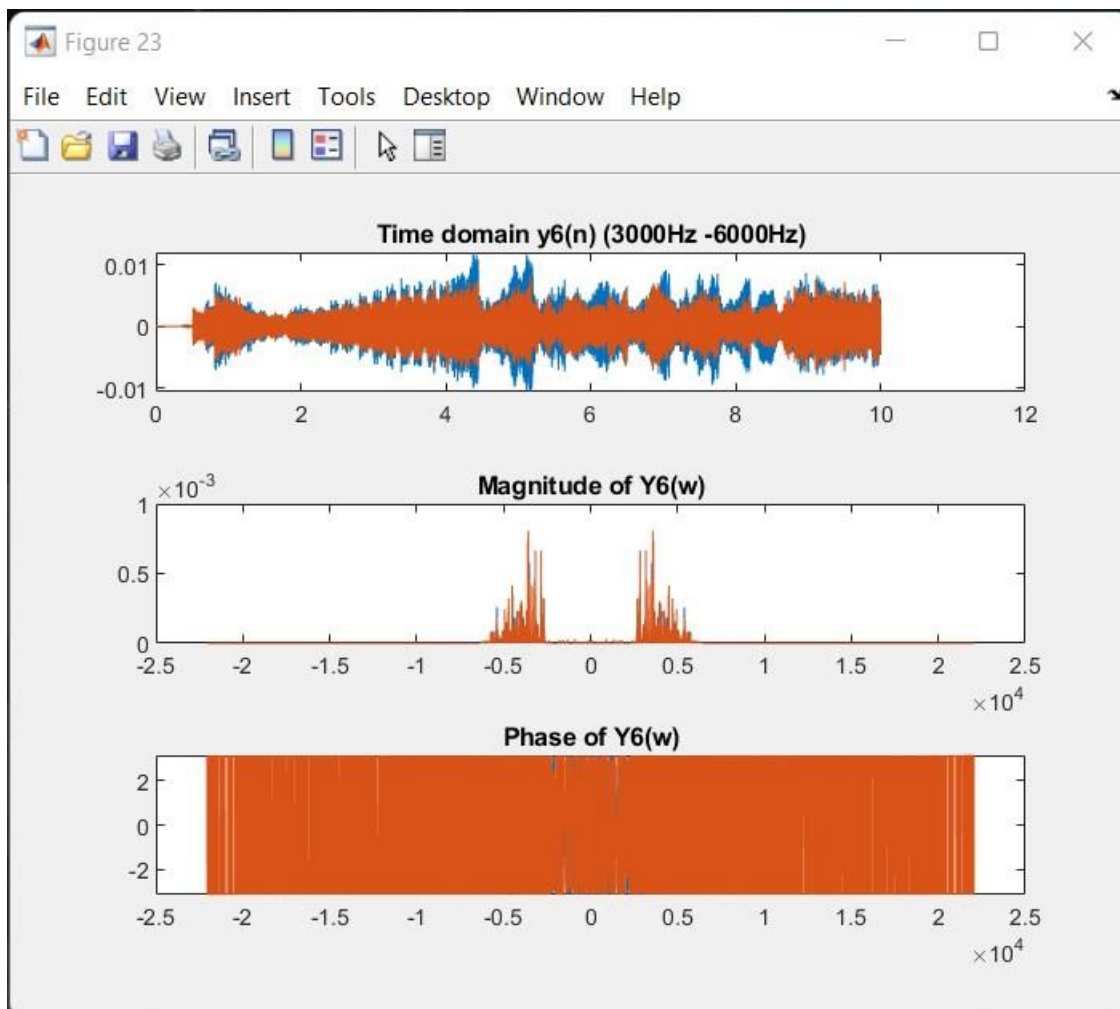
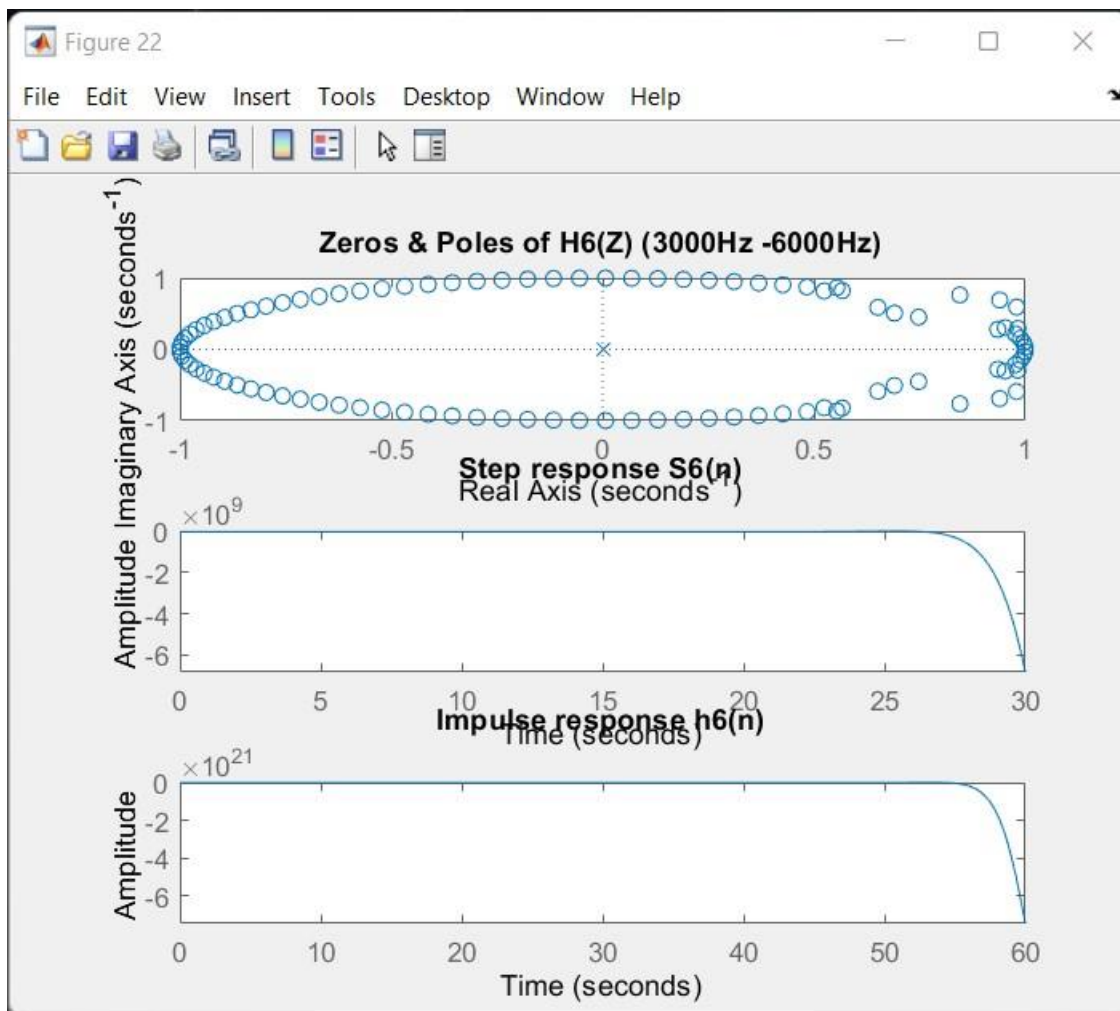


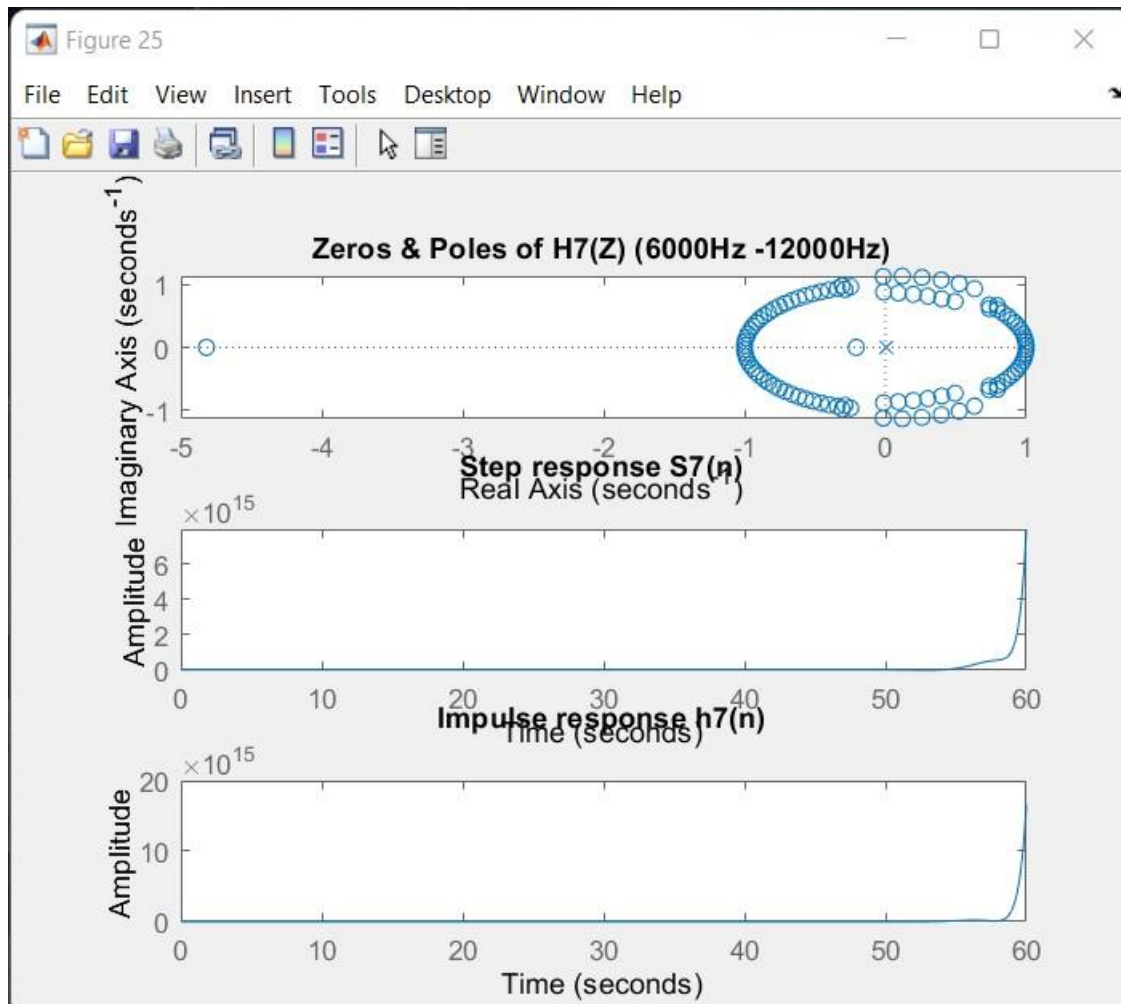
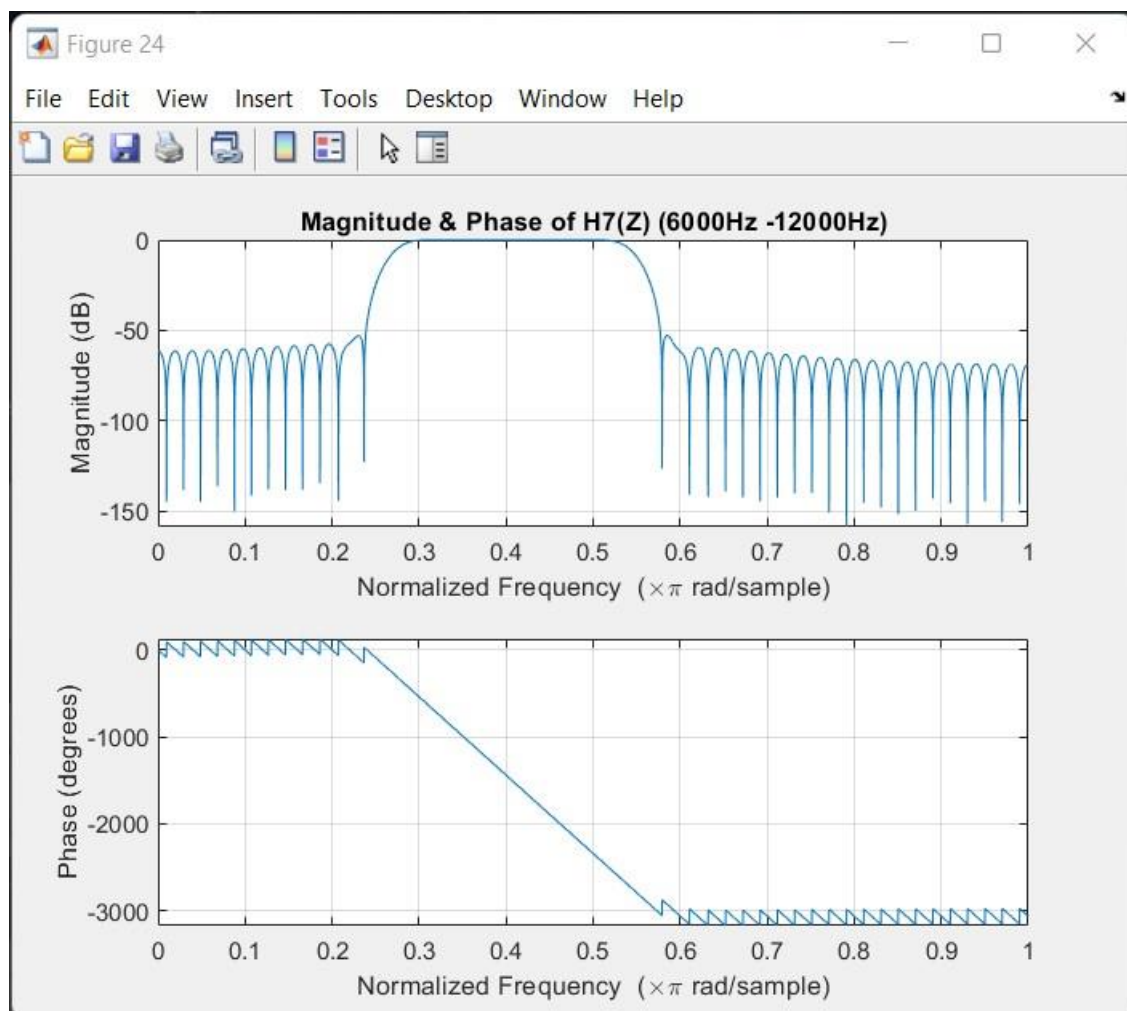


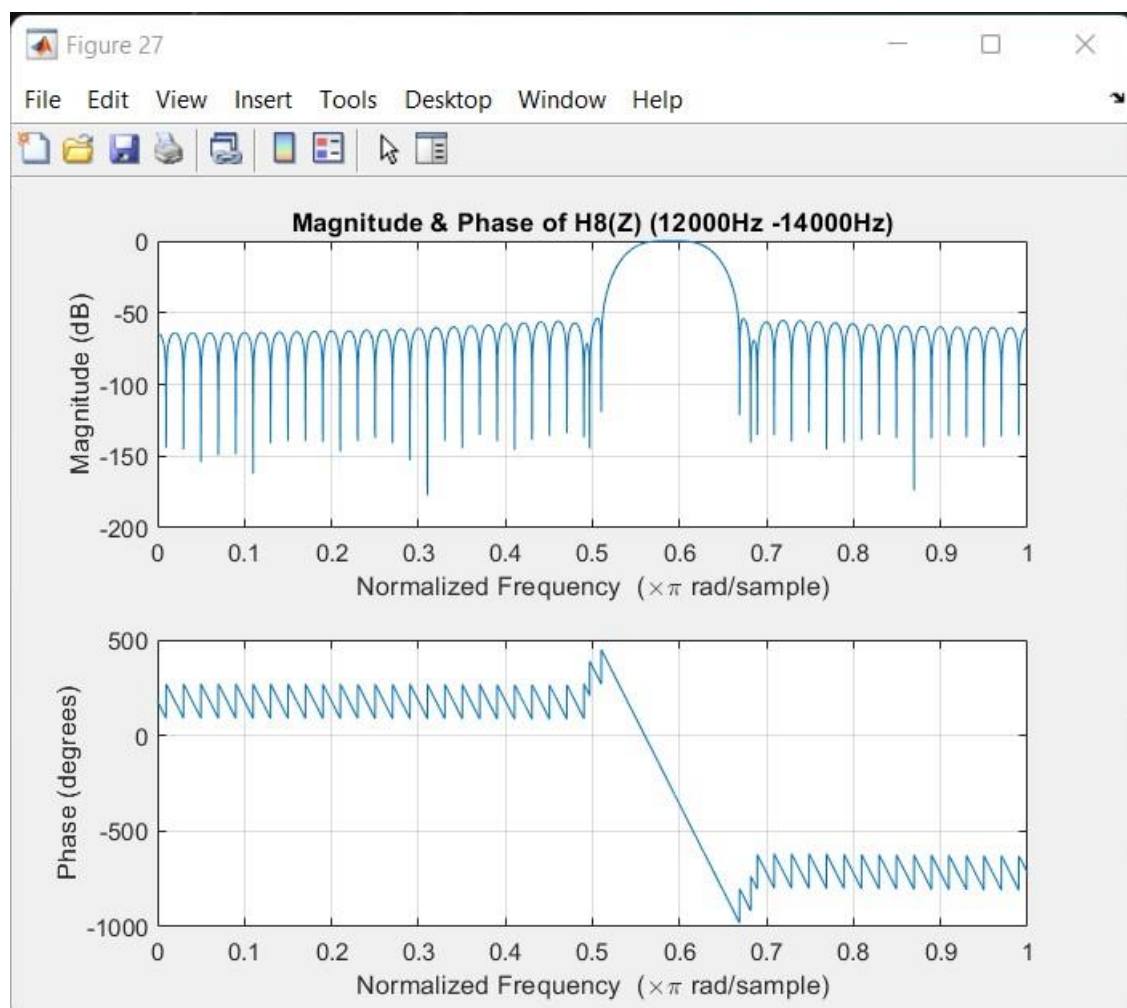
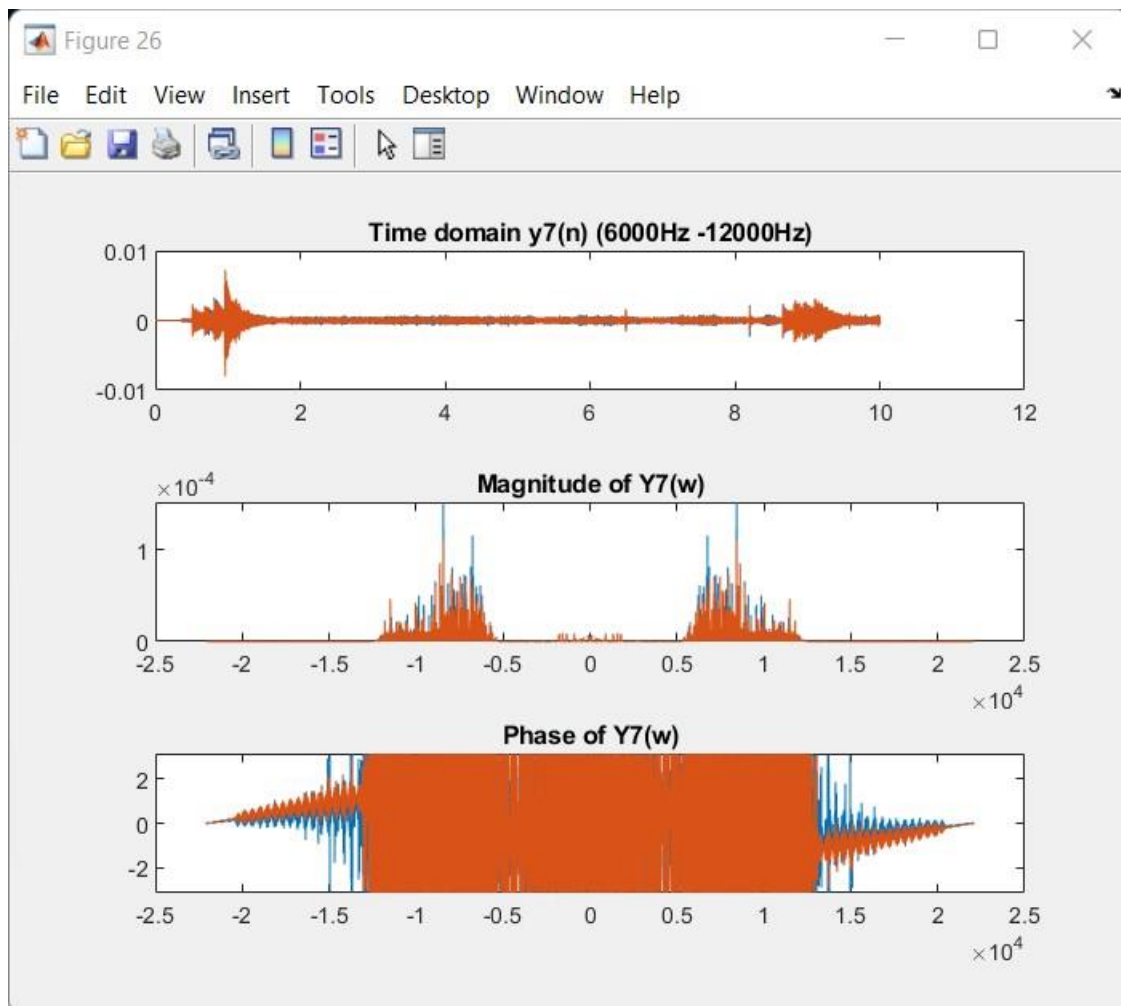


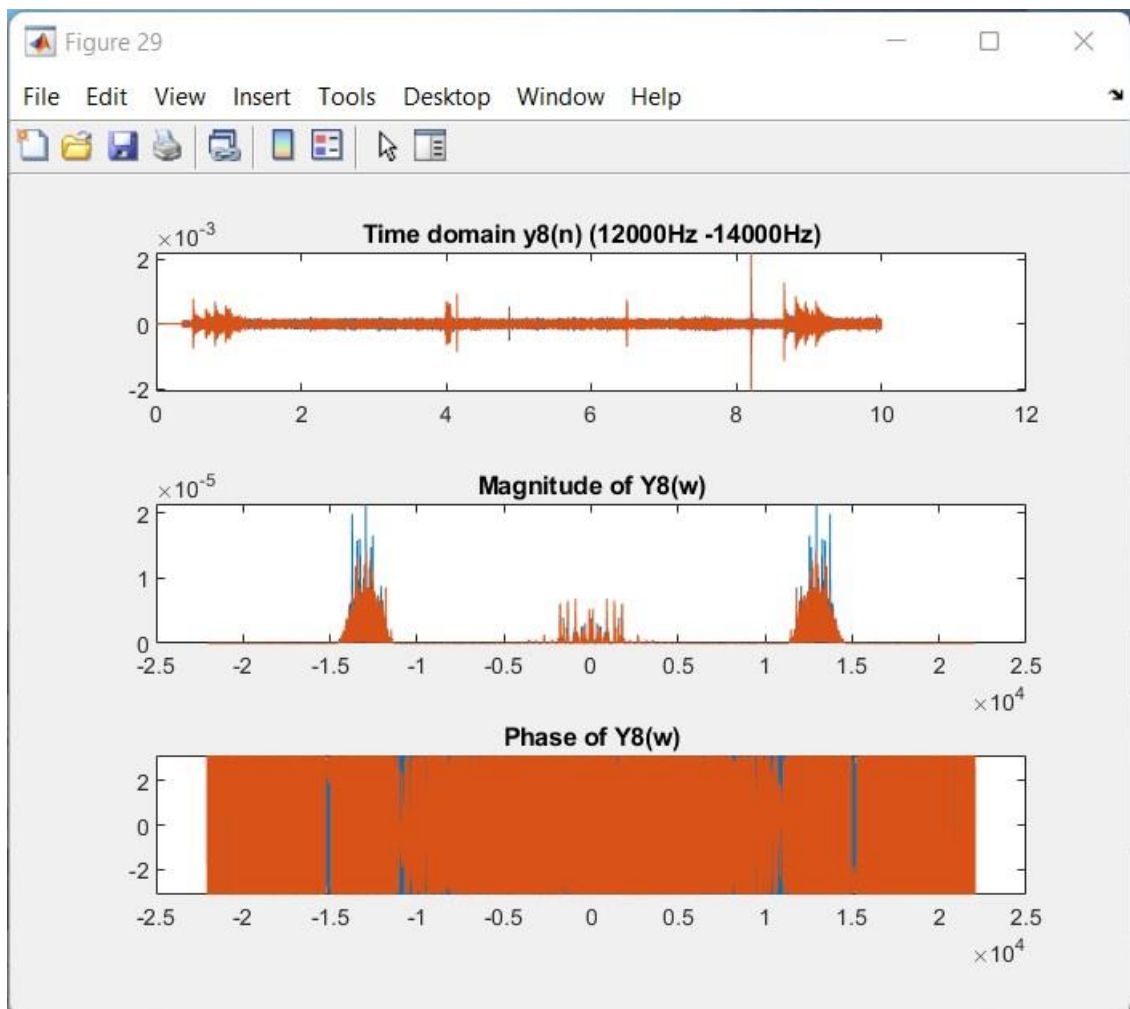
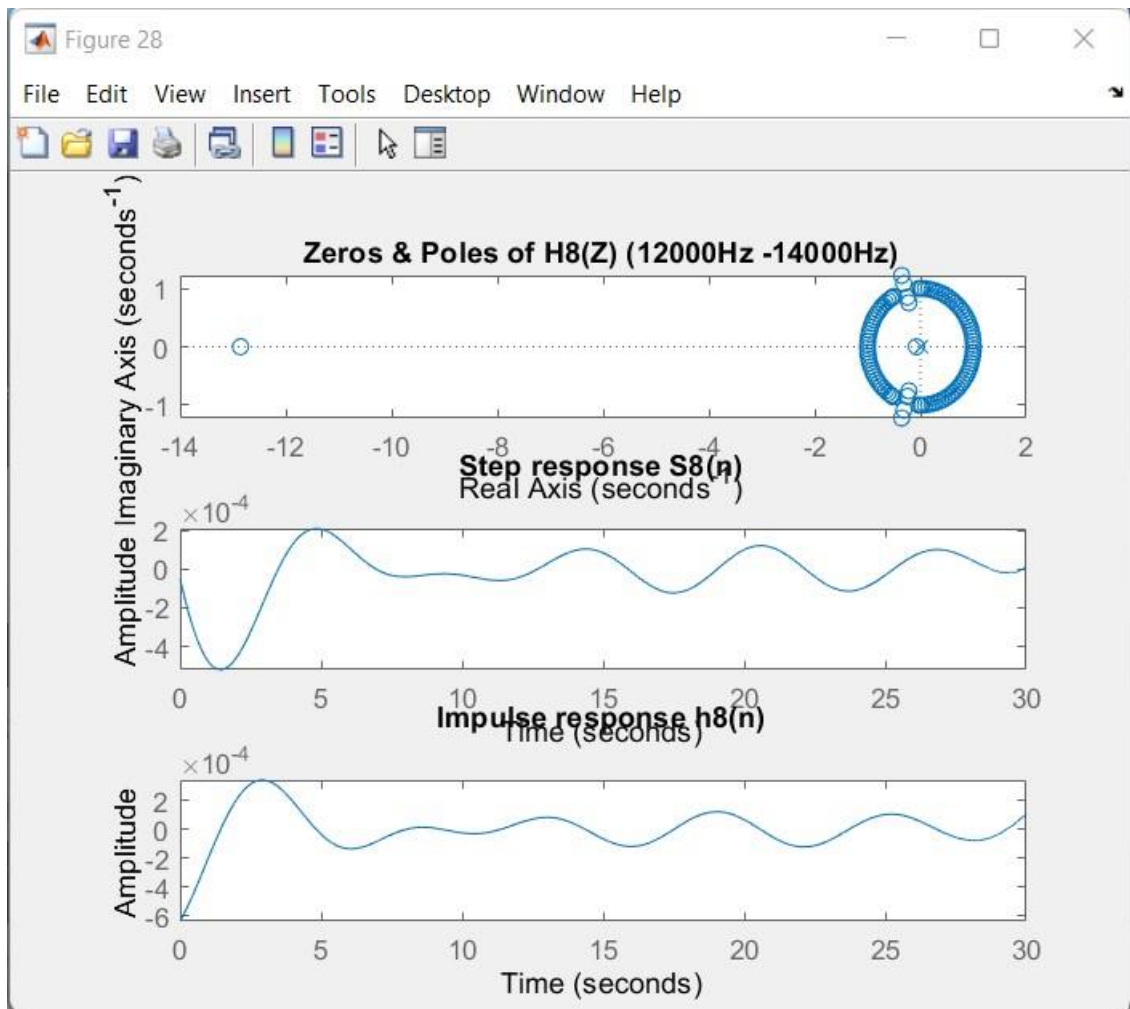


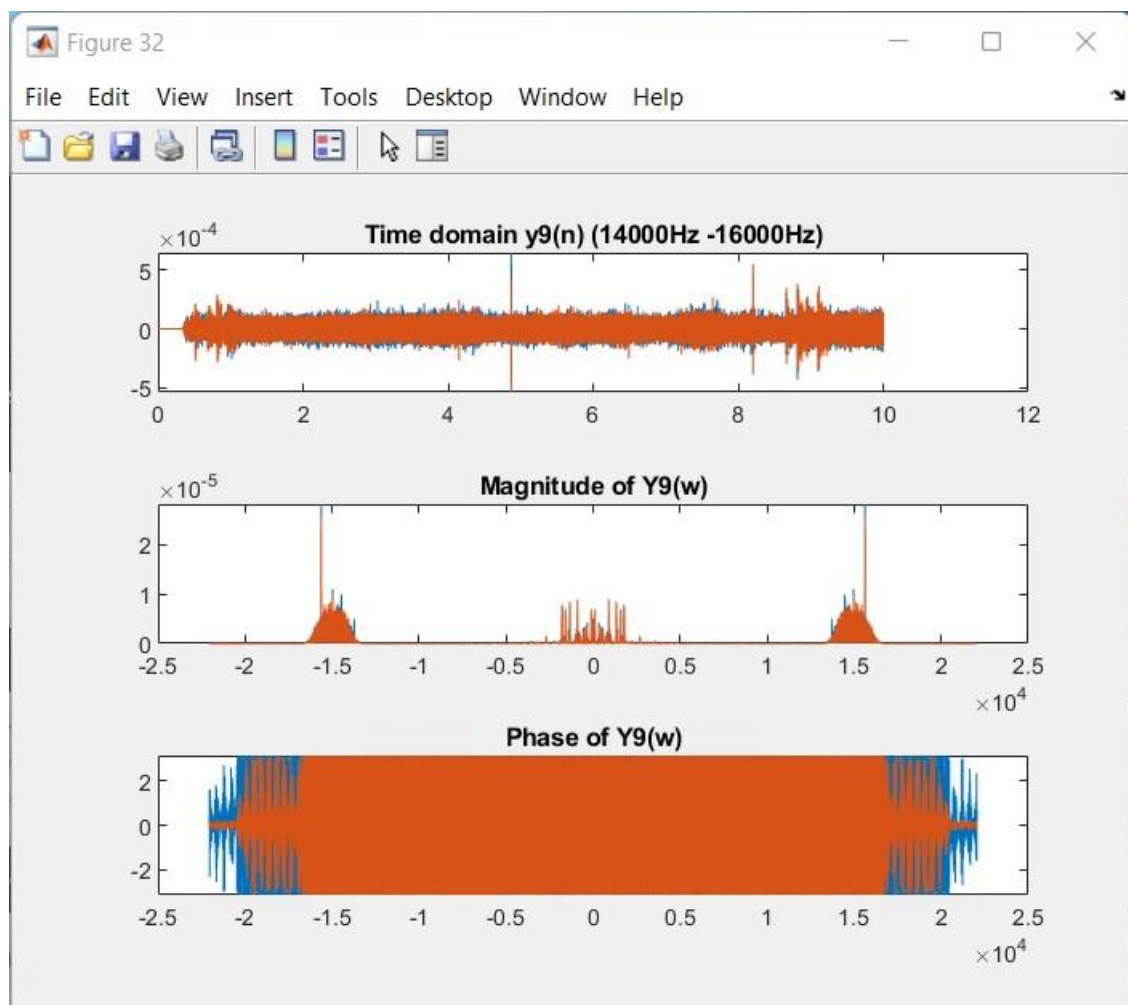
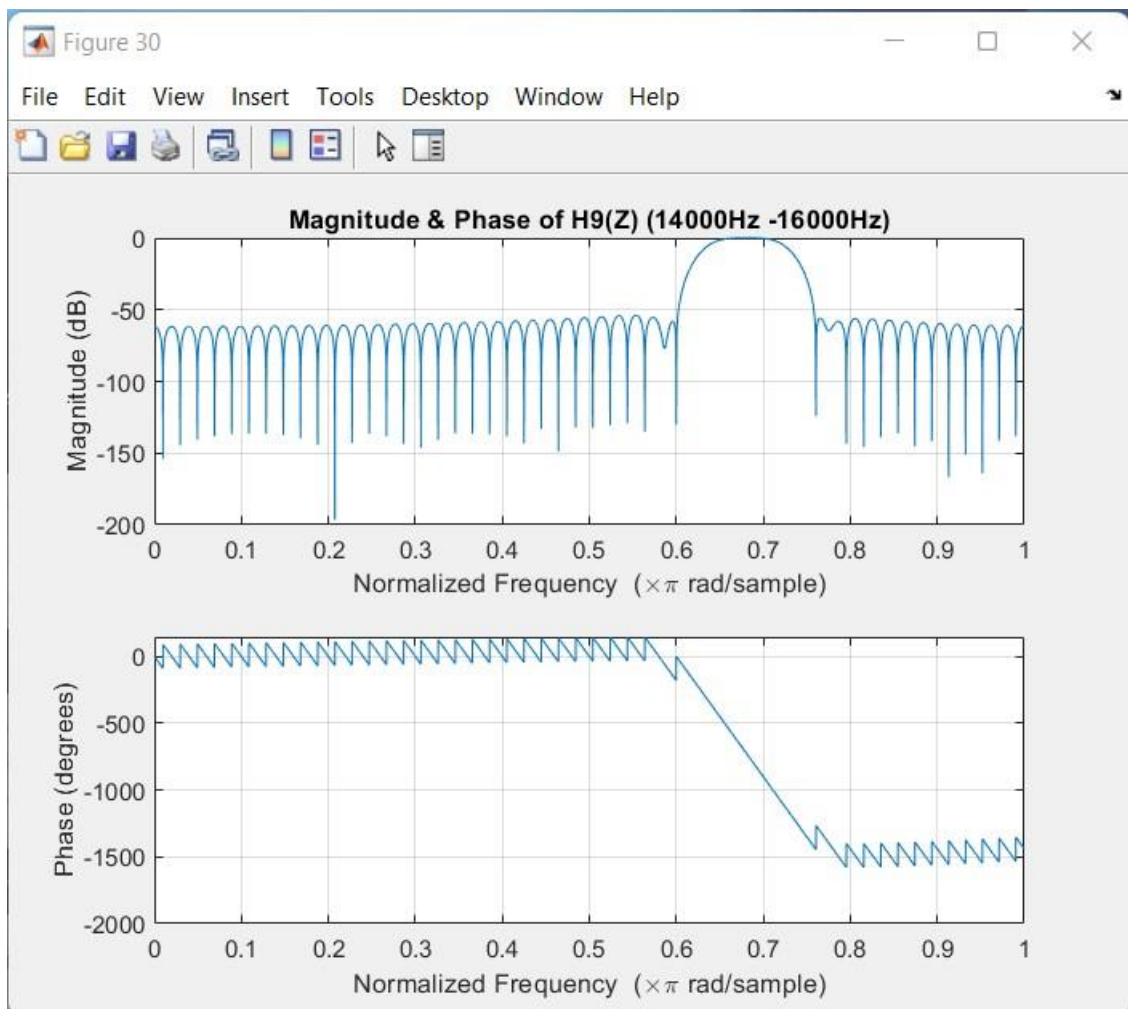


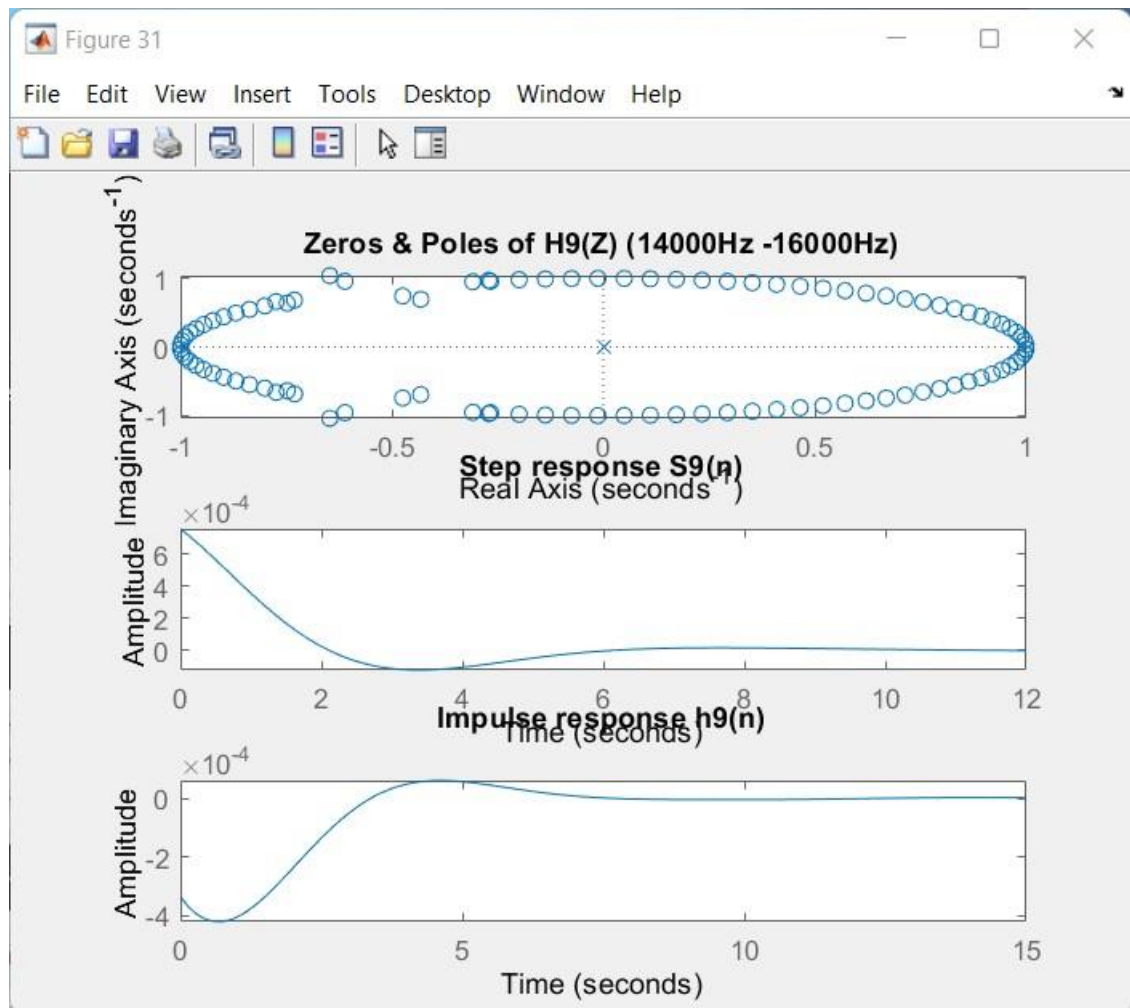




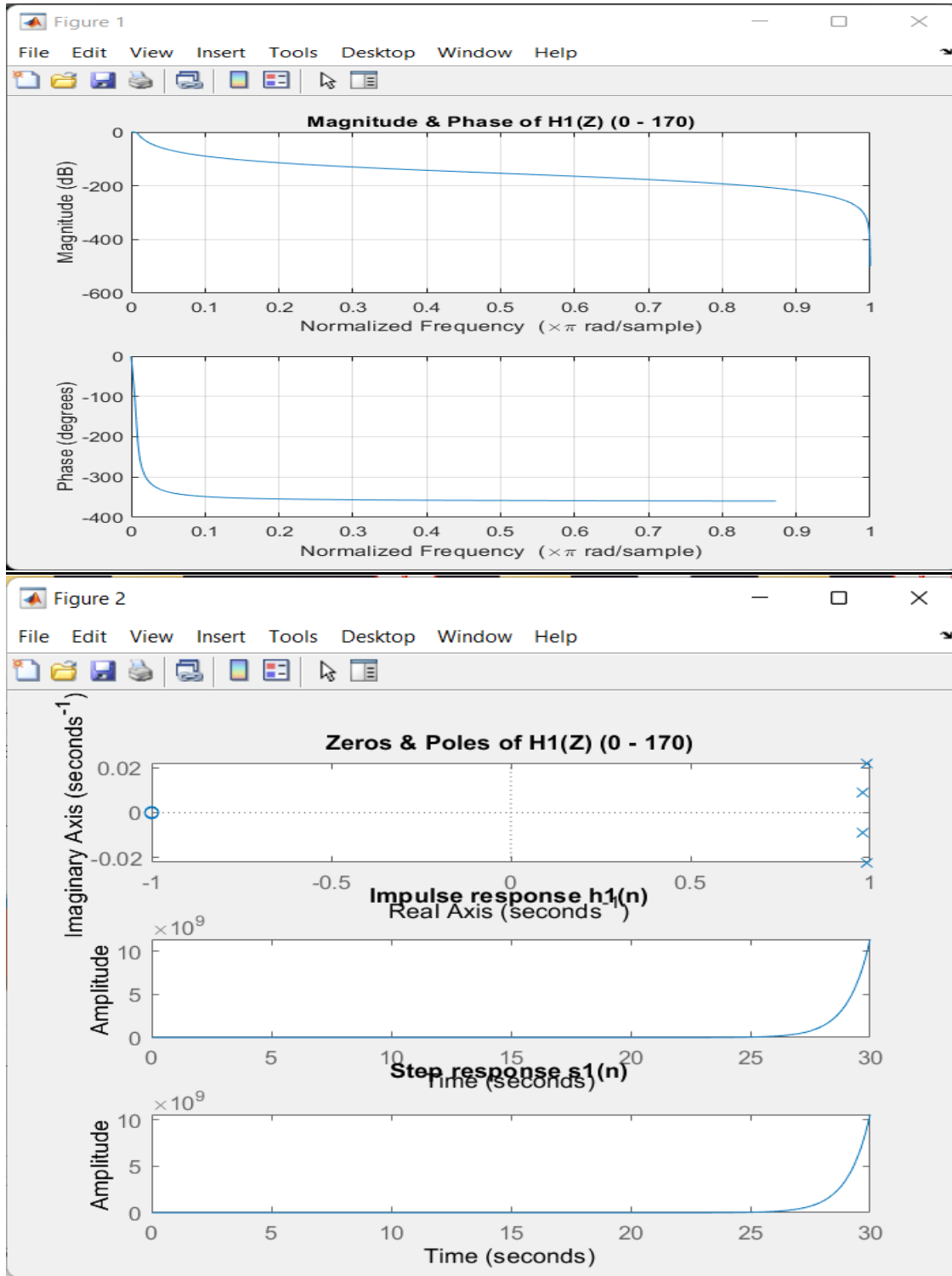


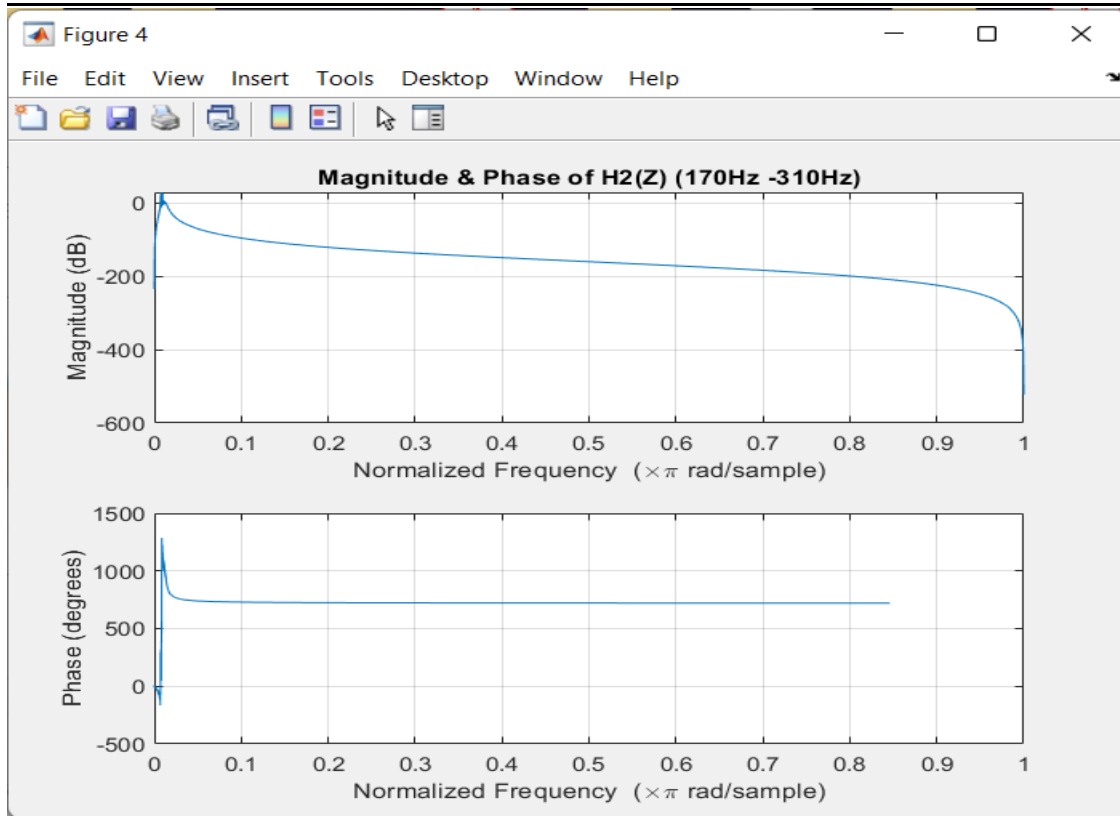
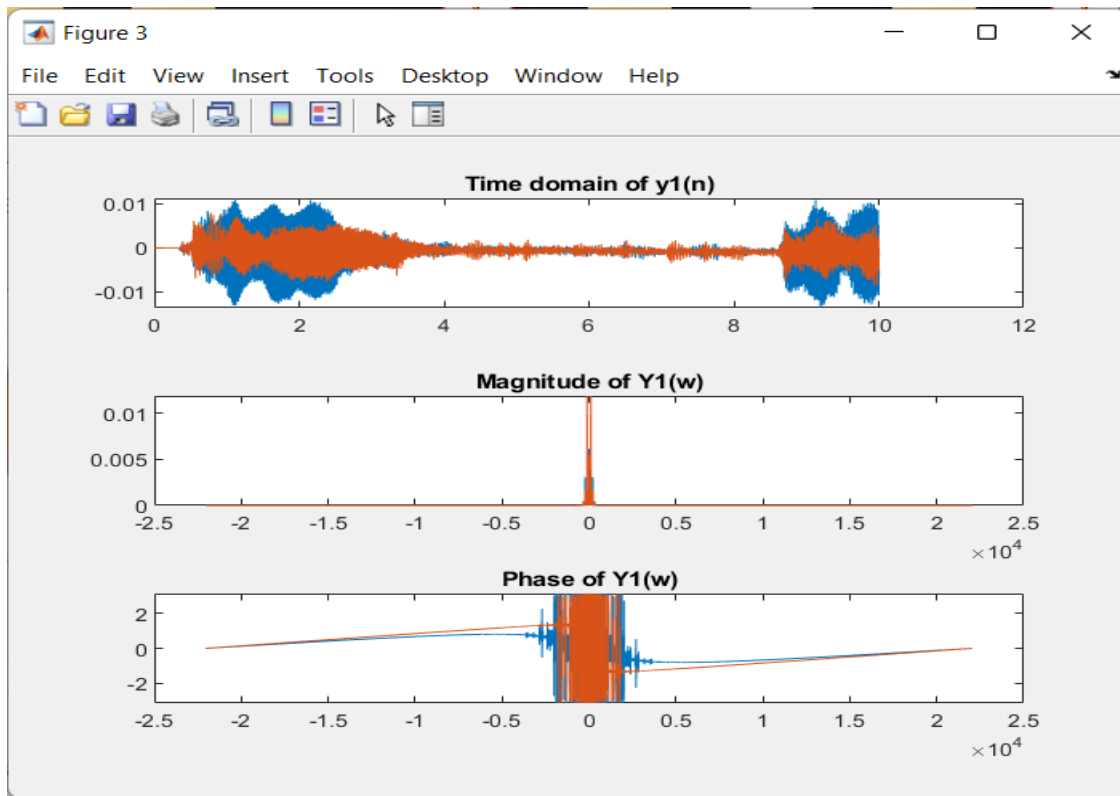


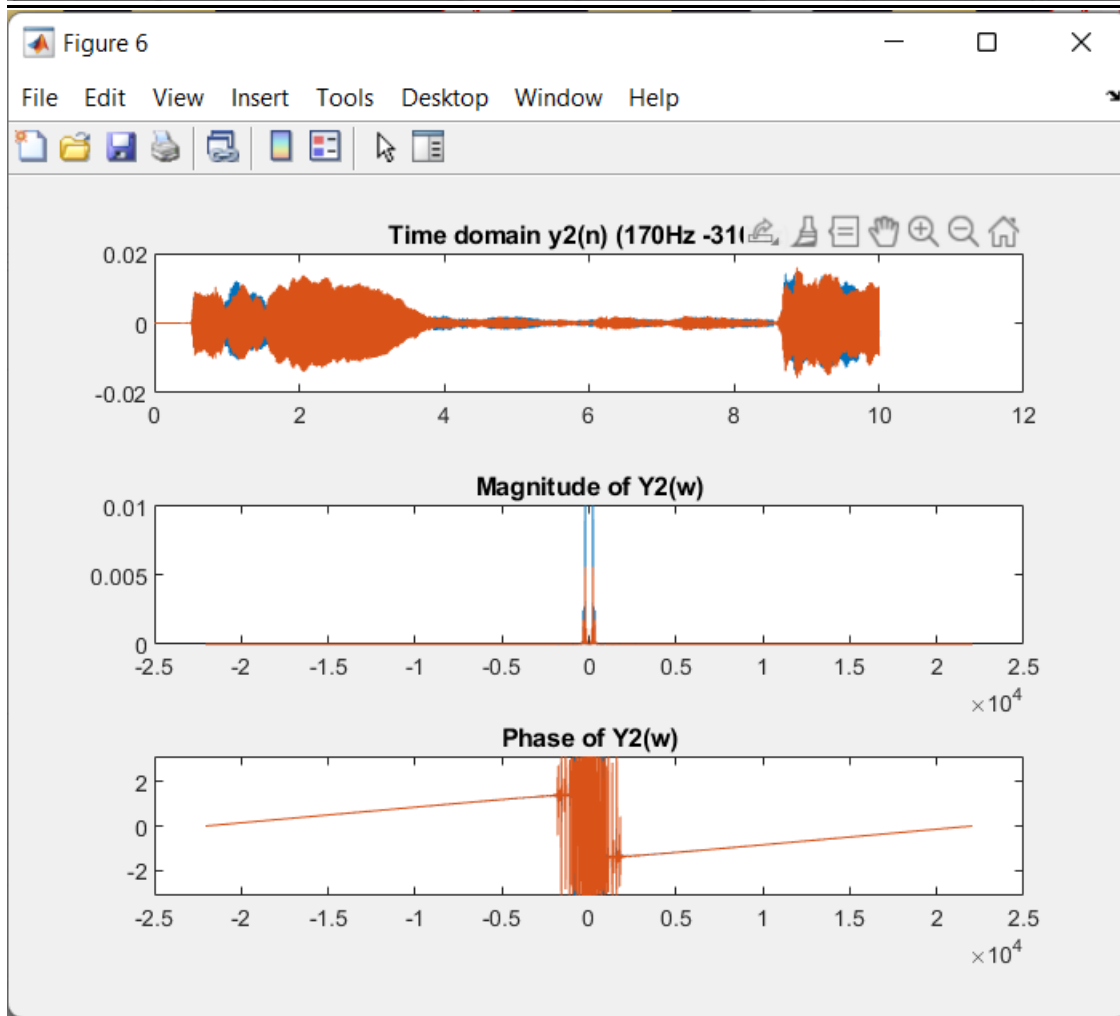
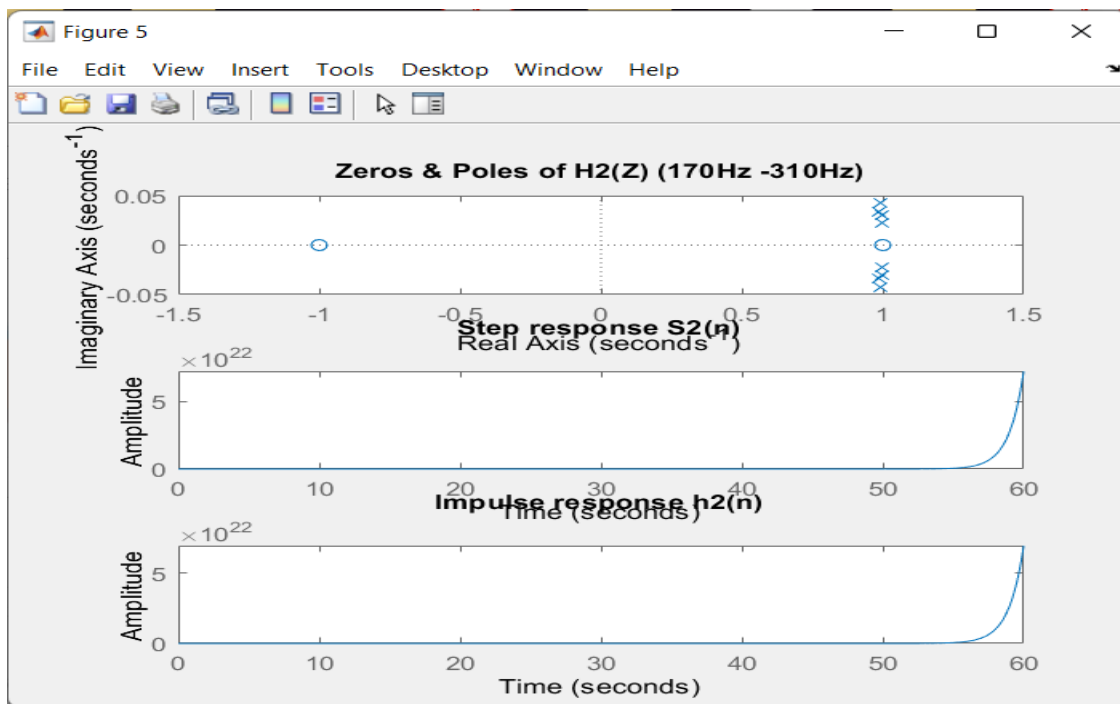


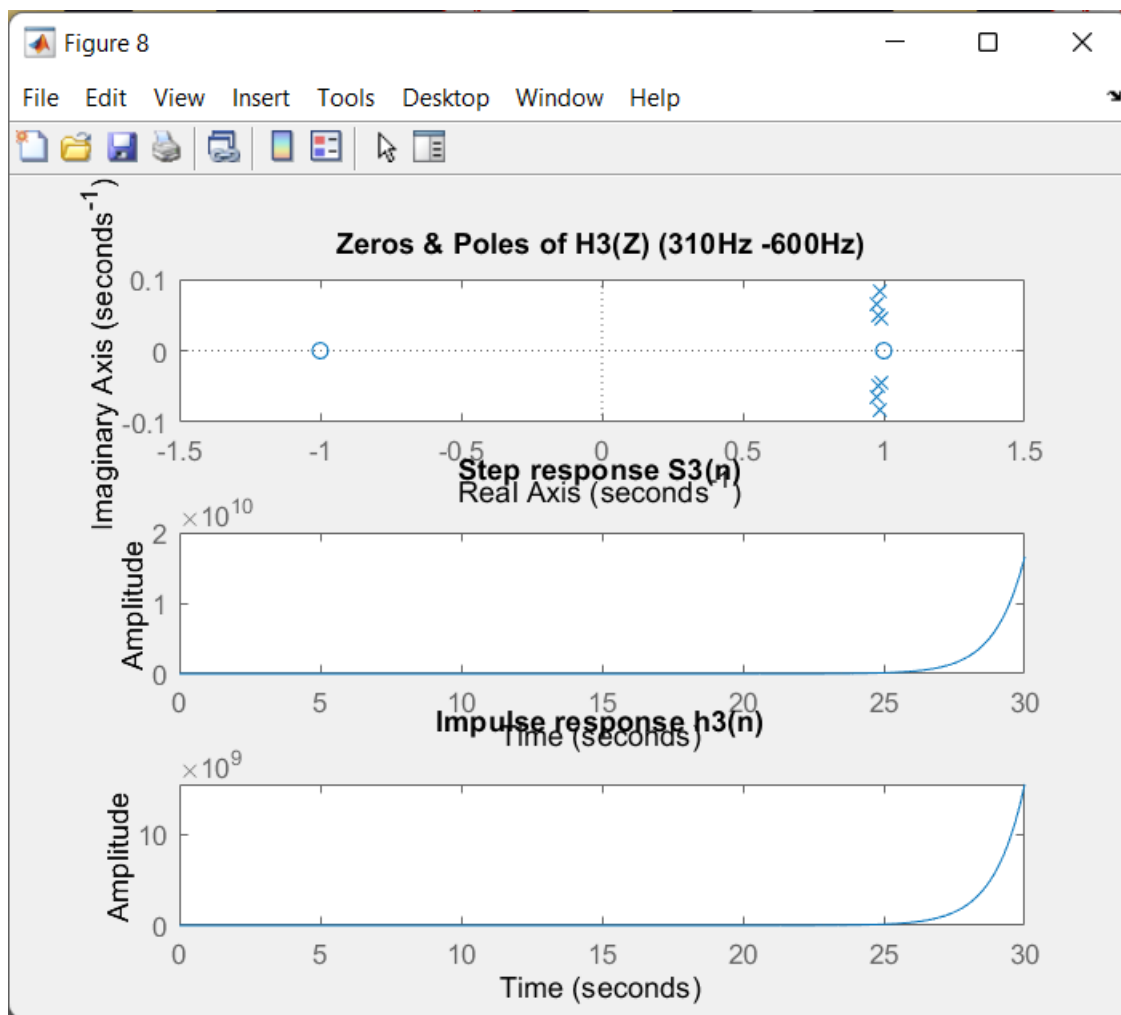
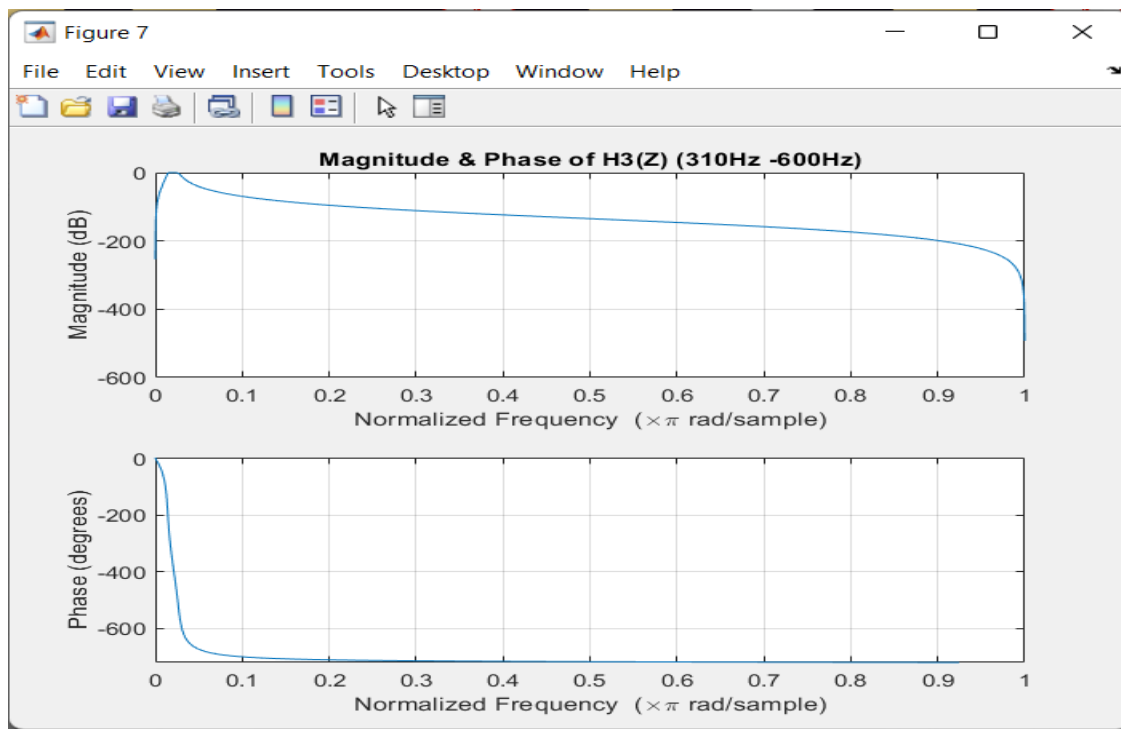


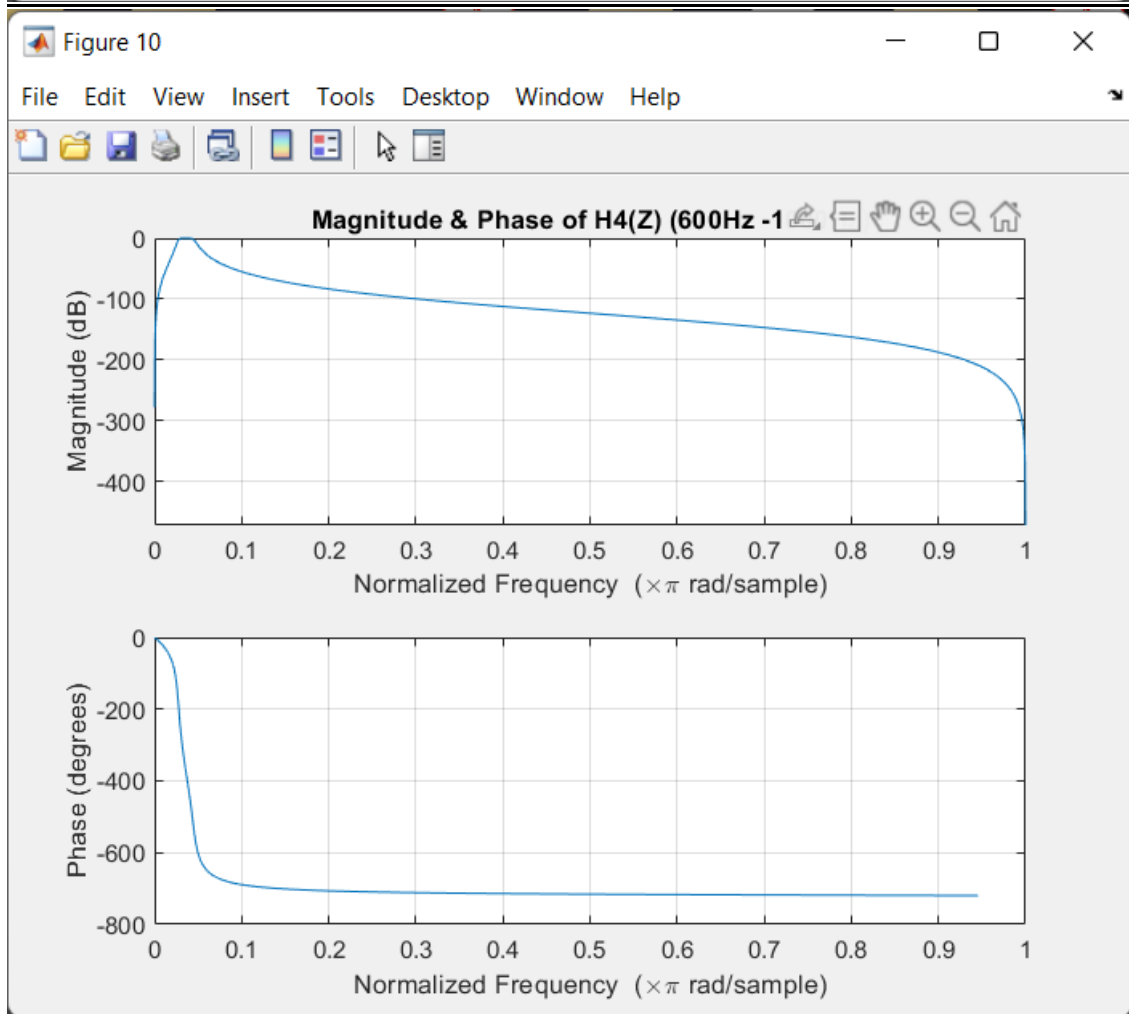
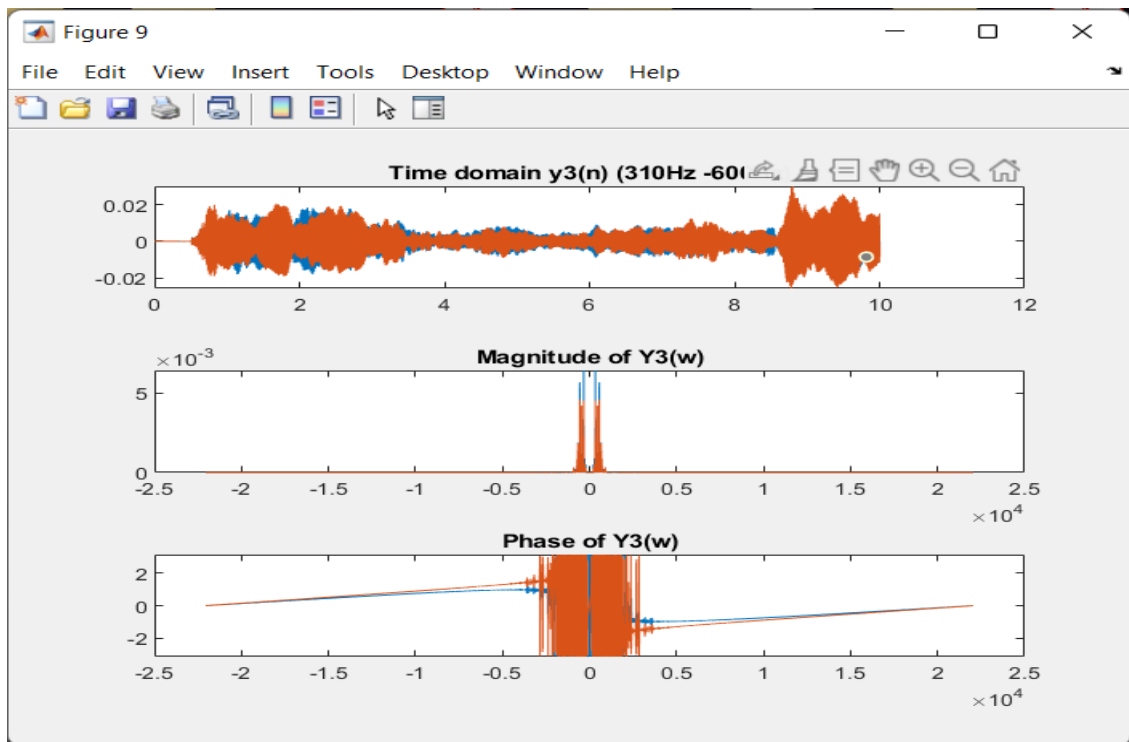
IIR Analysis

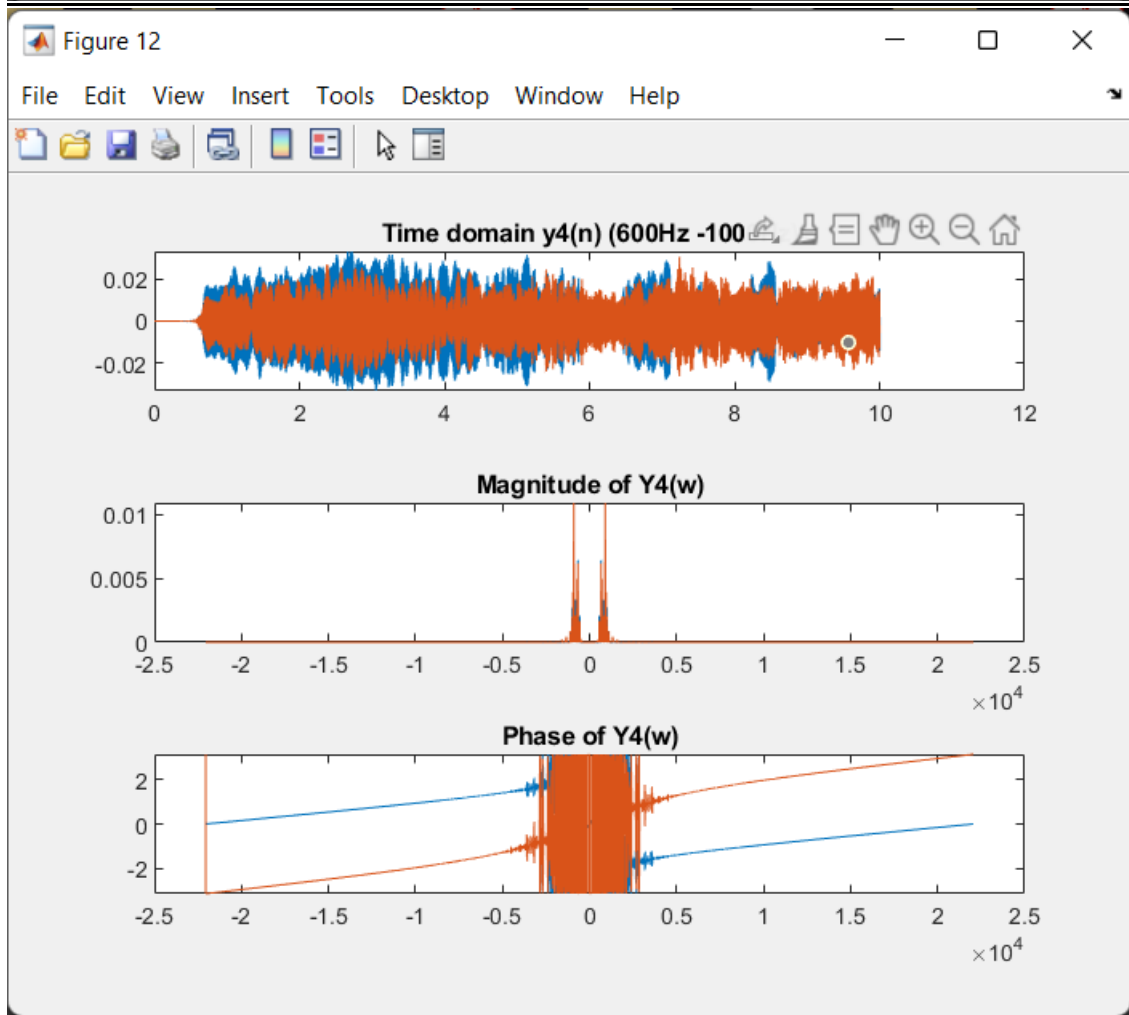
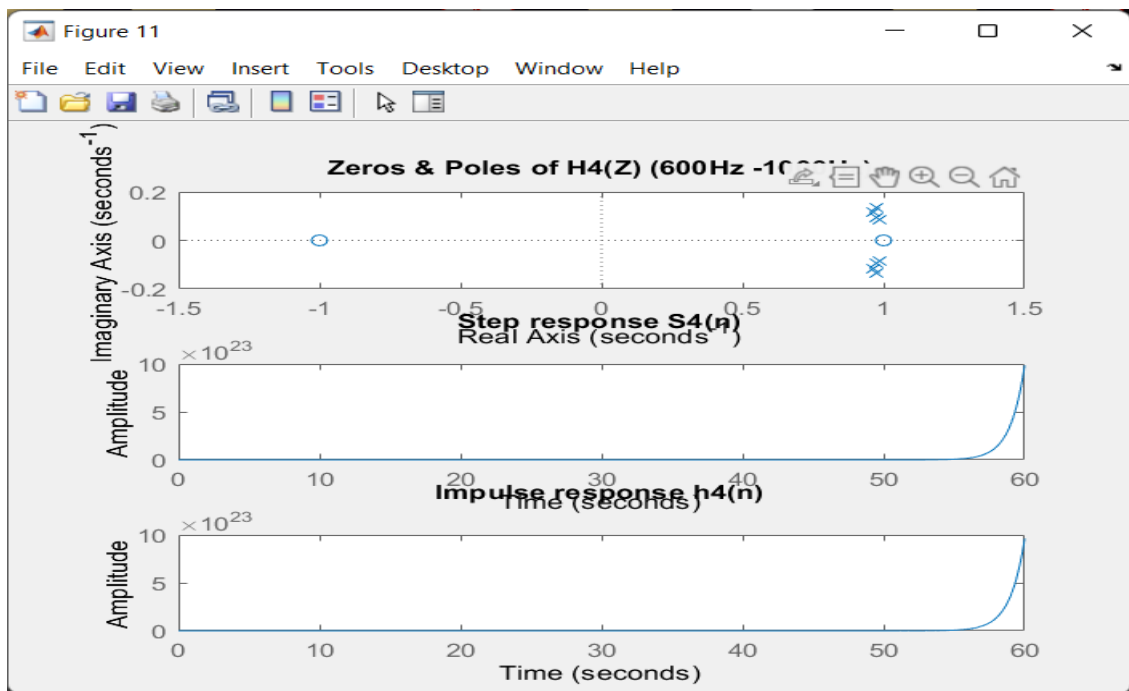


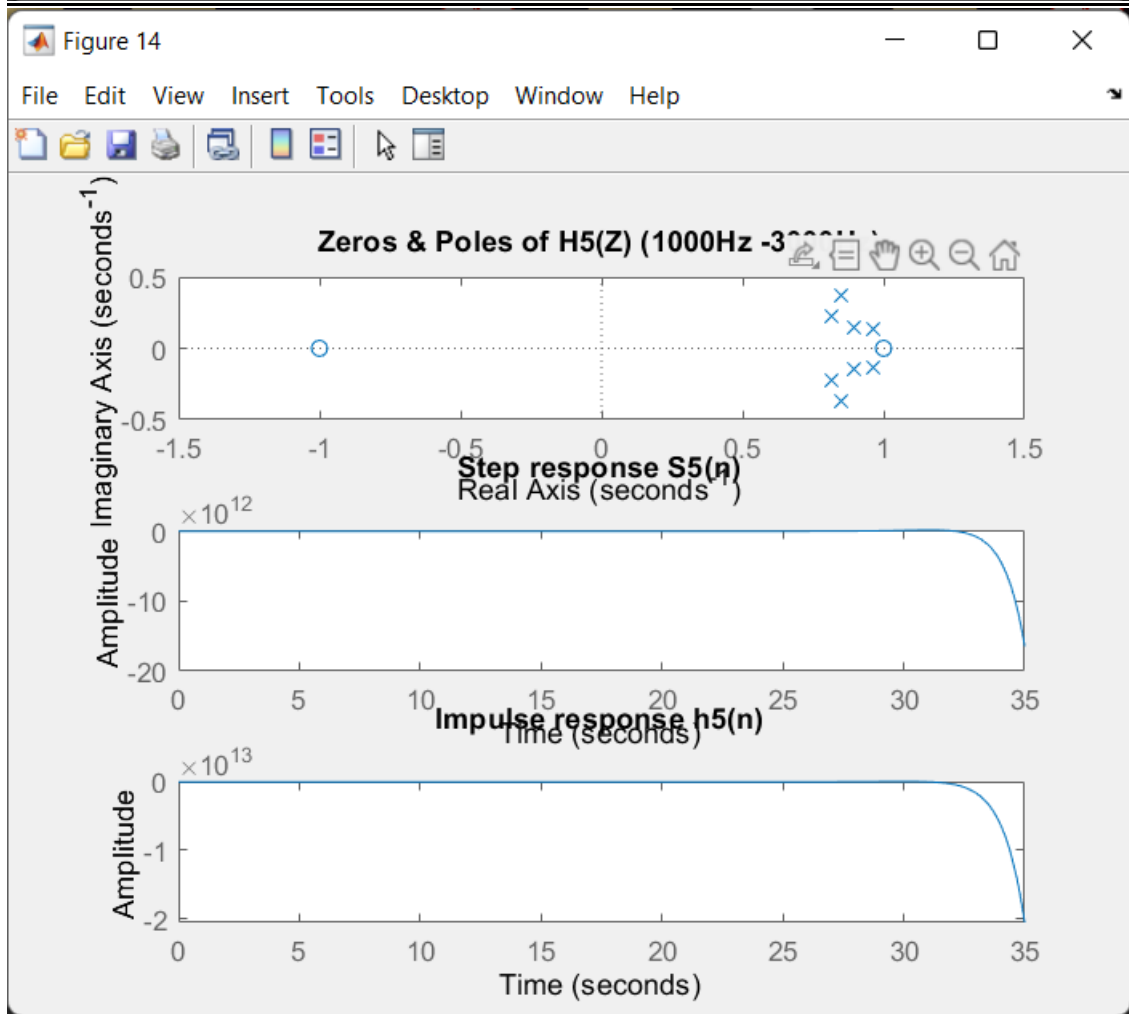
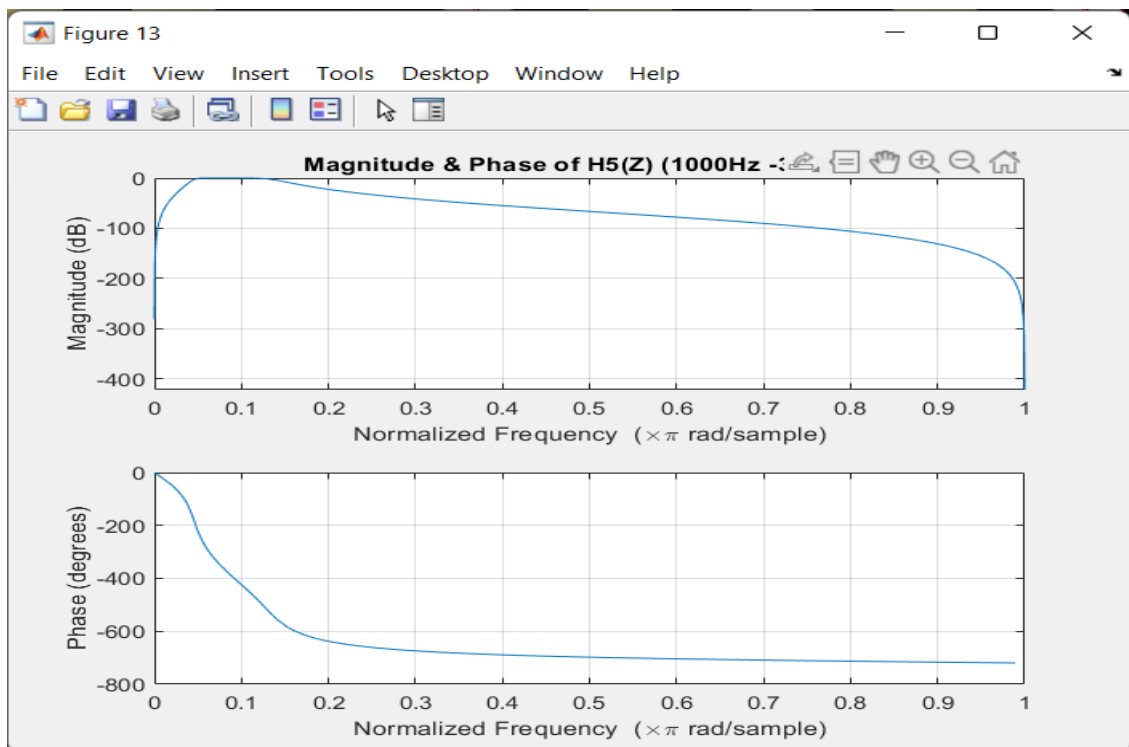


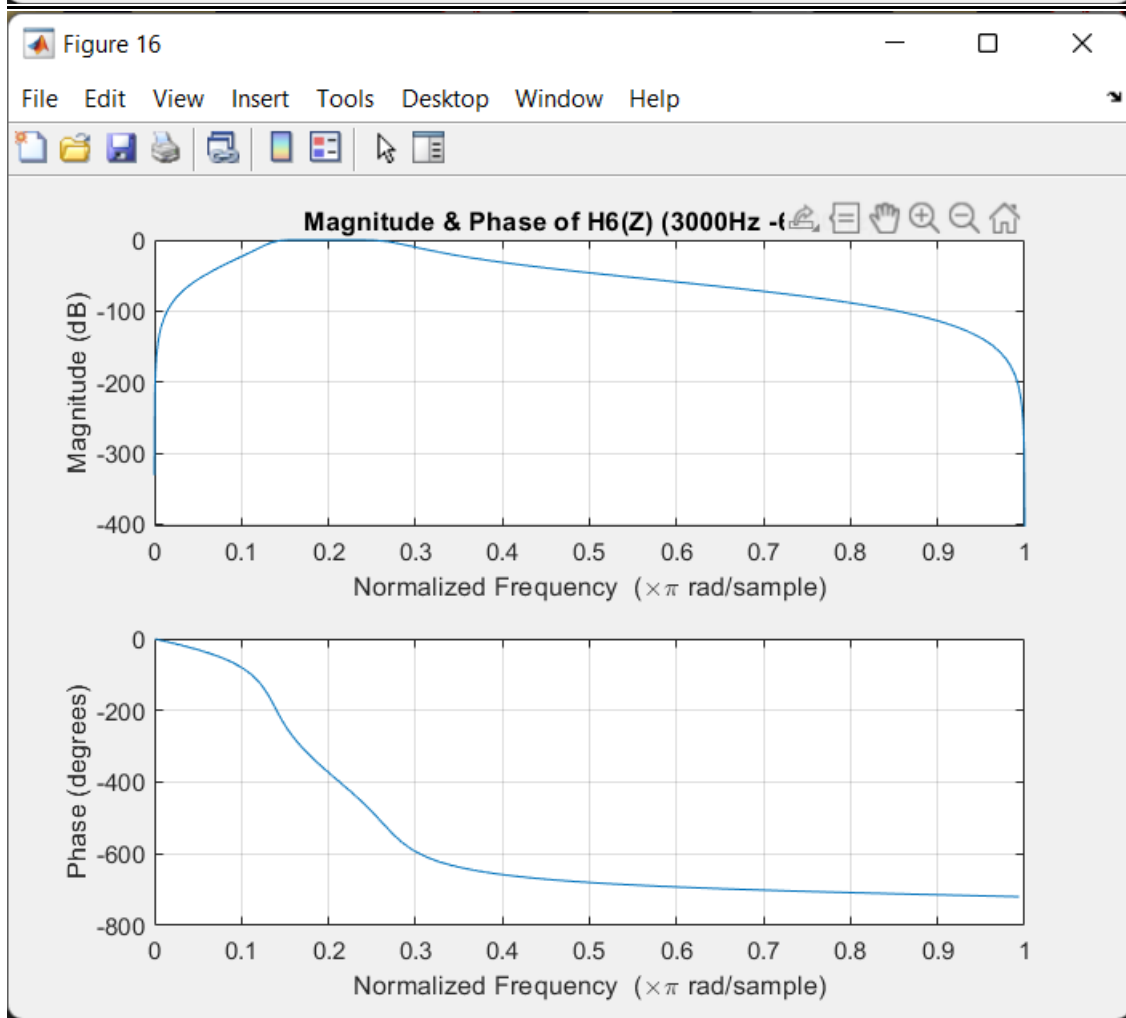
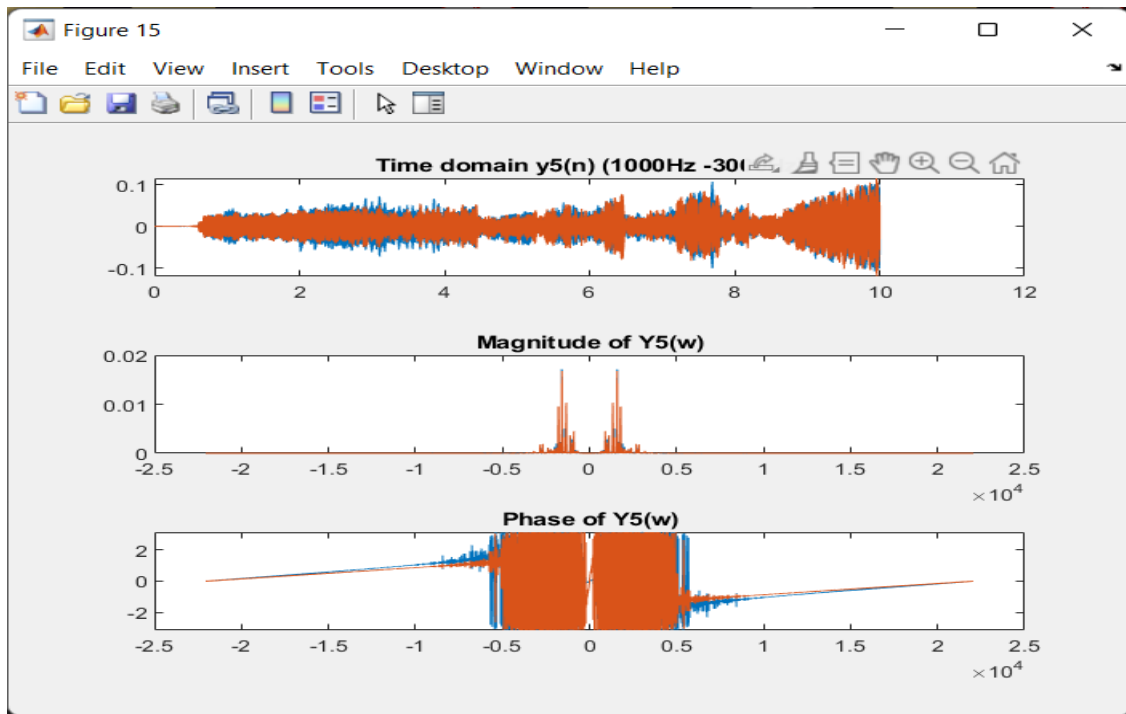


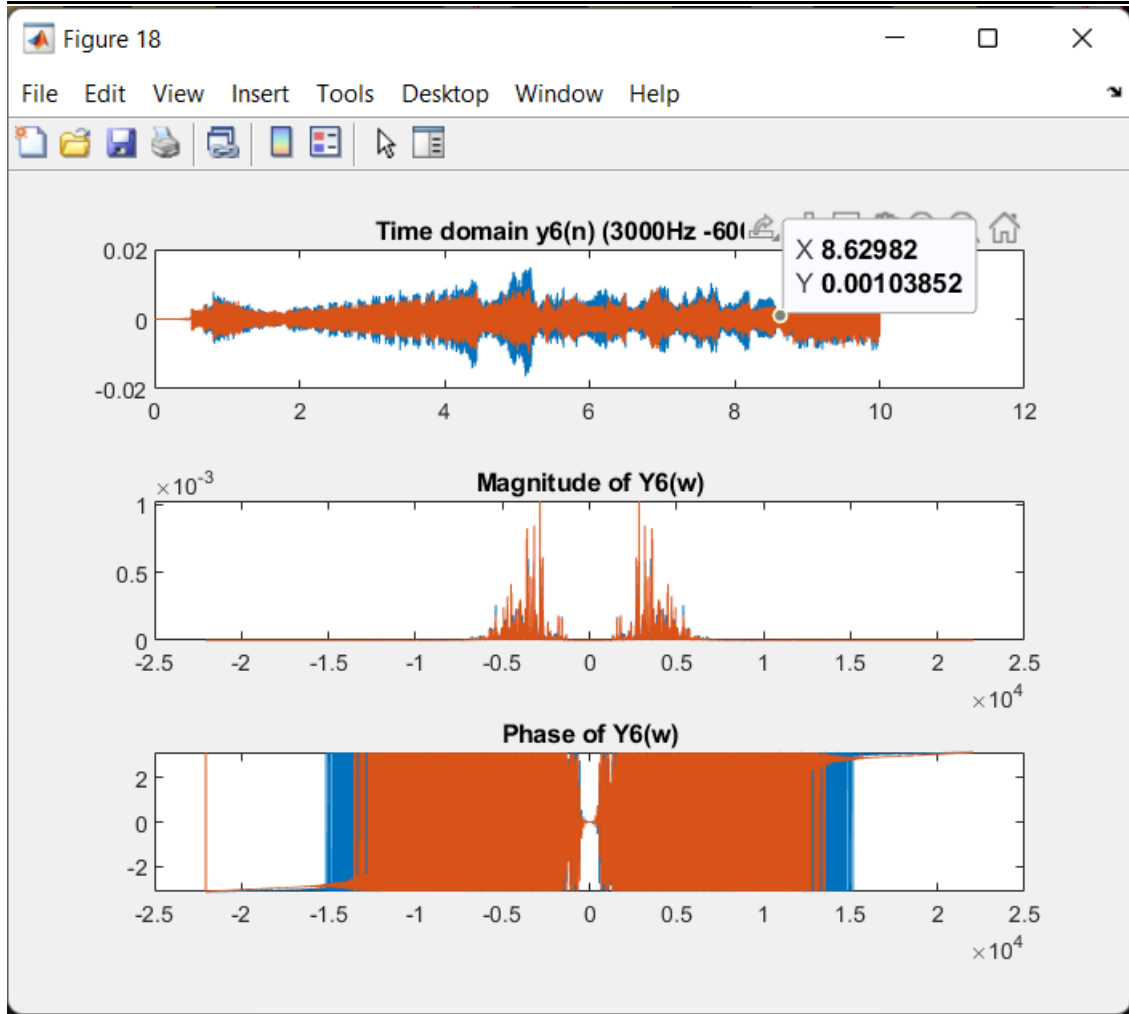
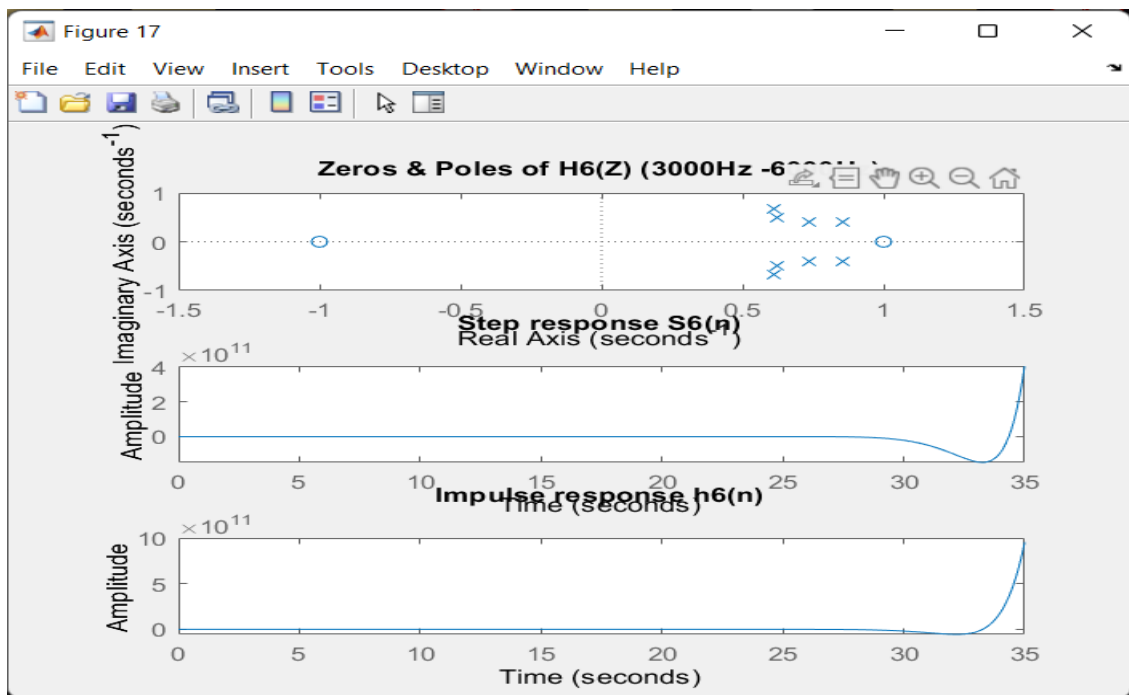


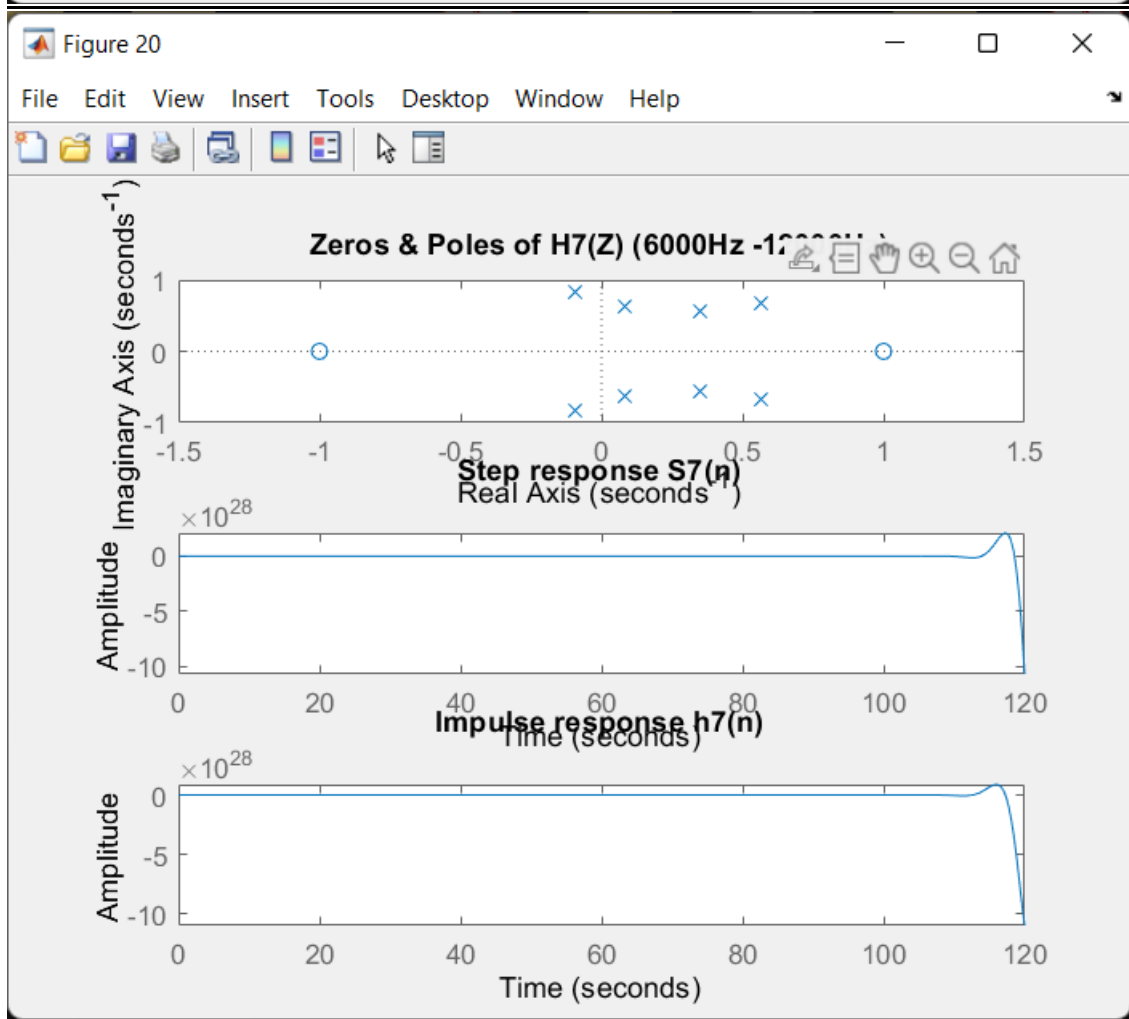
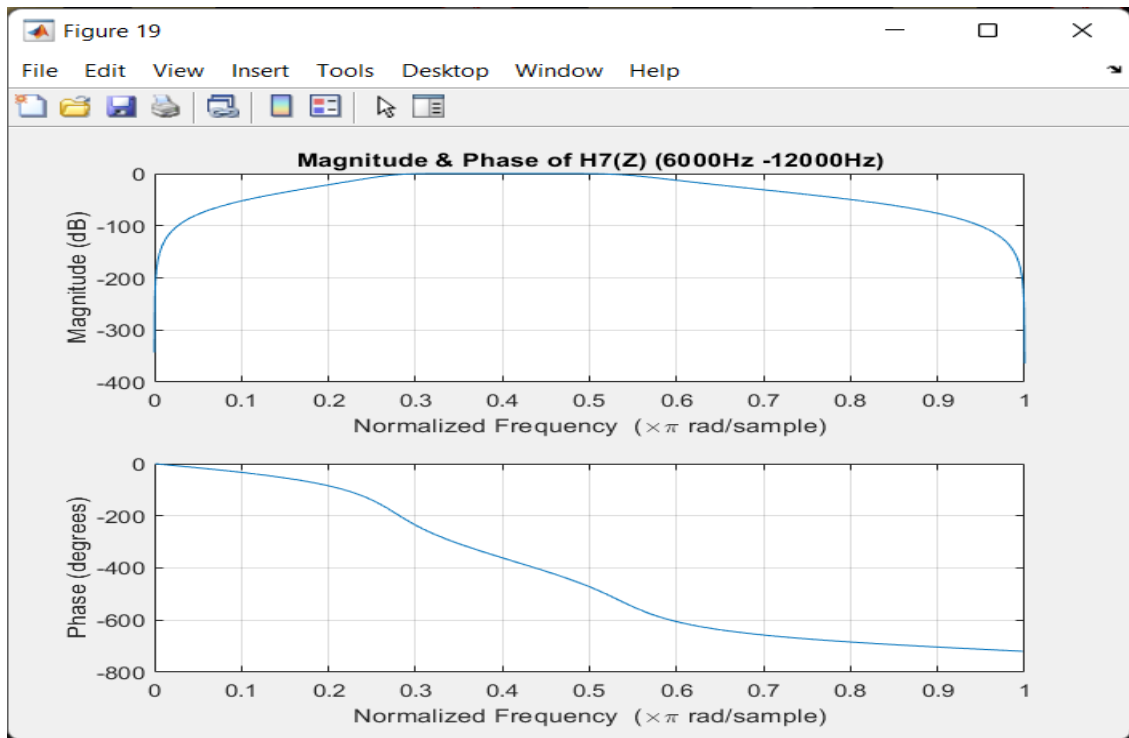


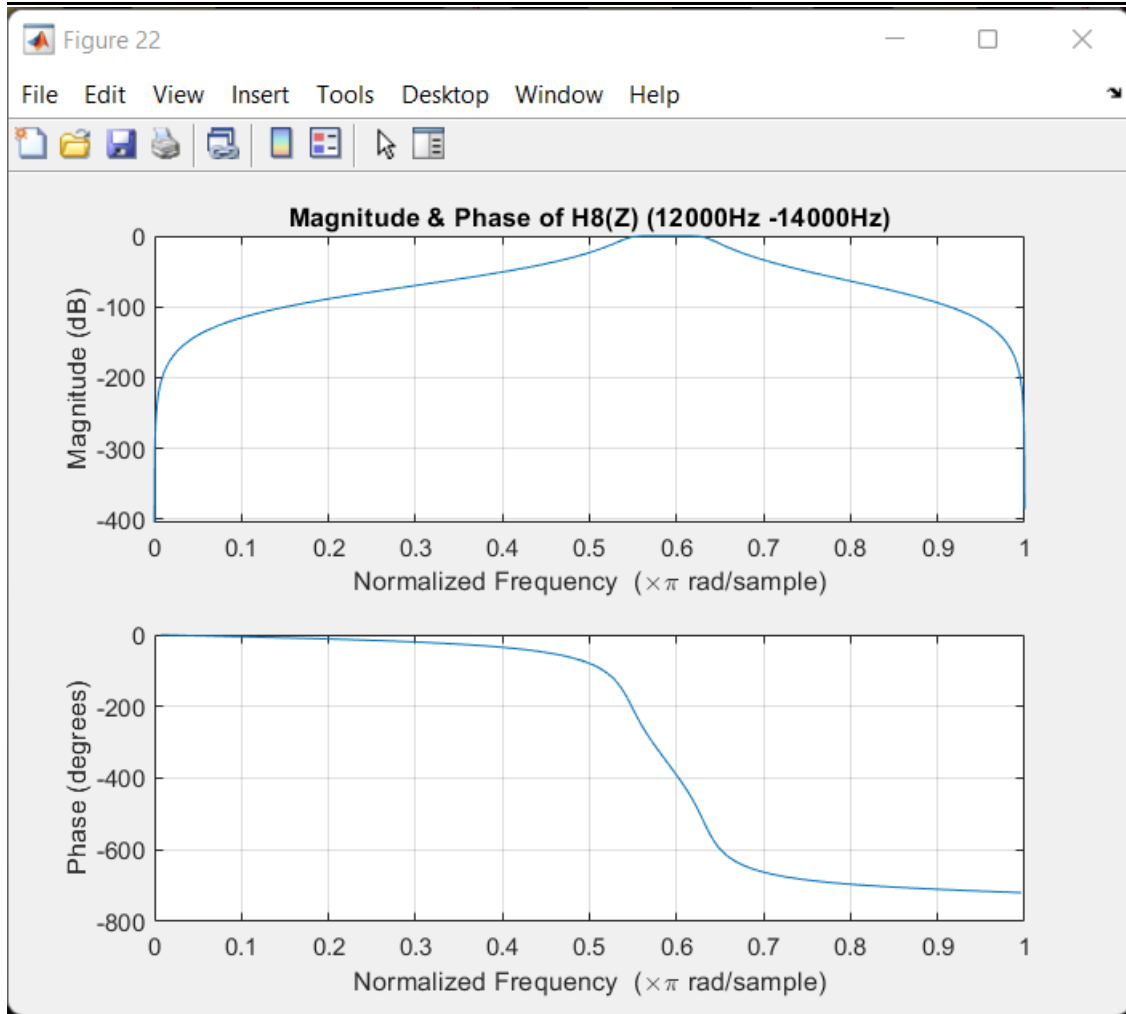
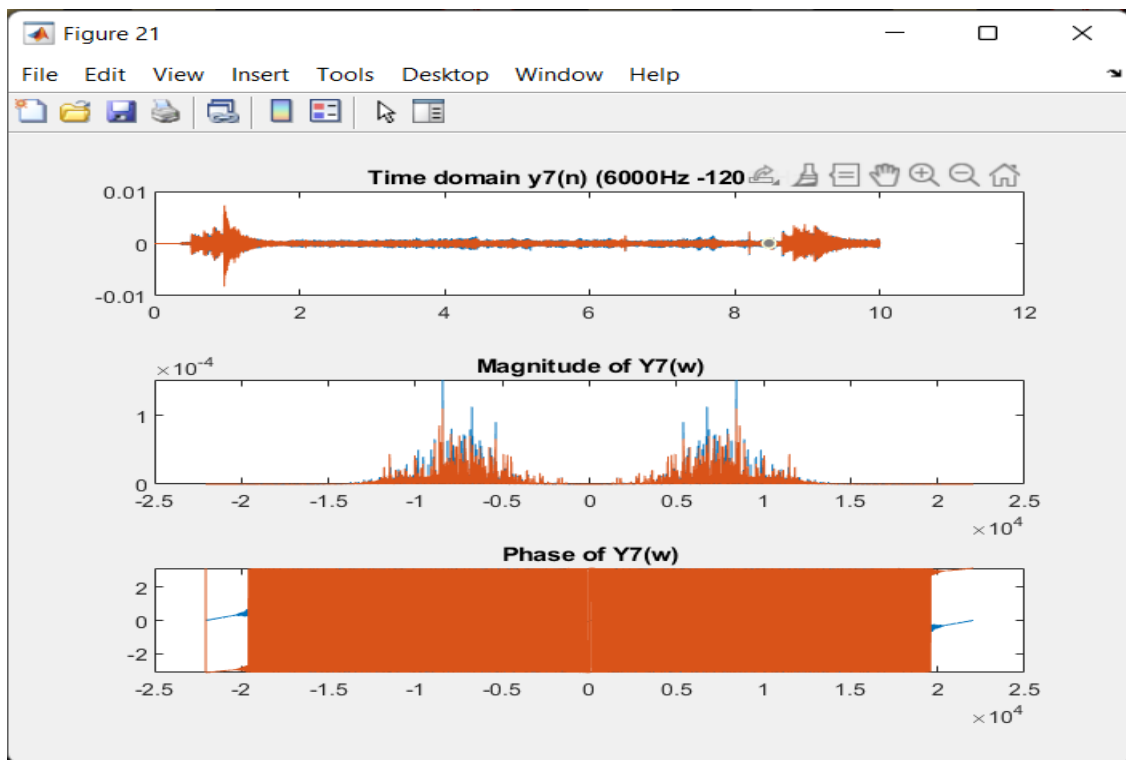


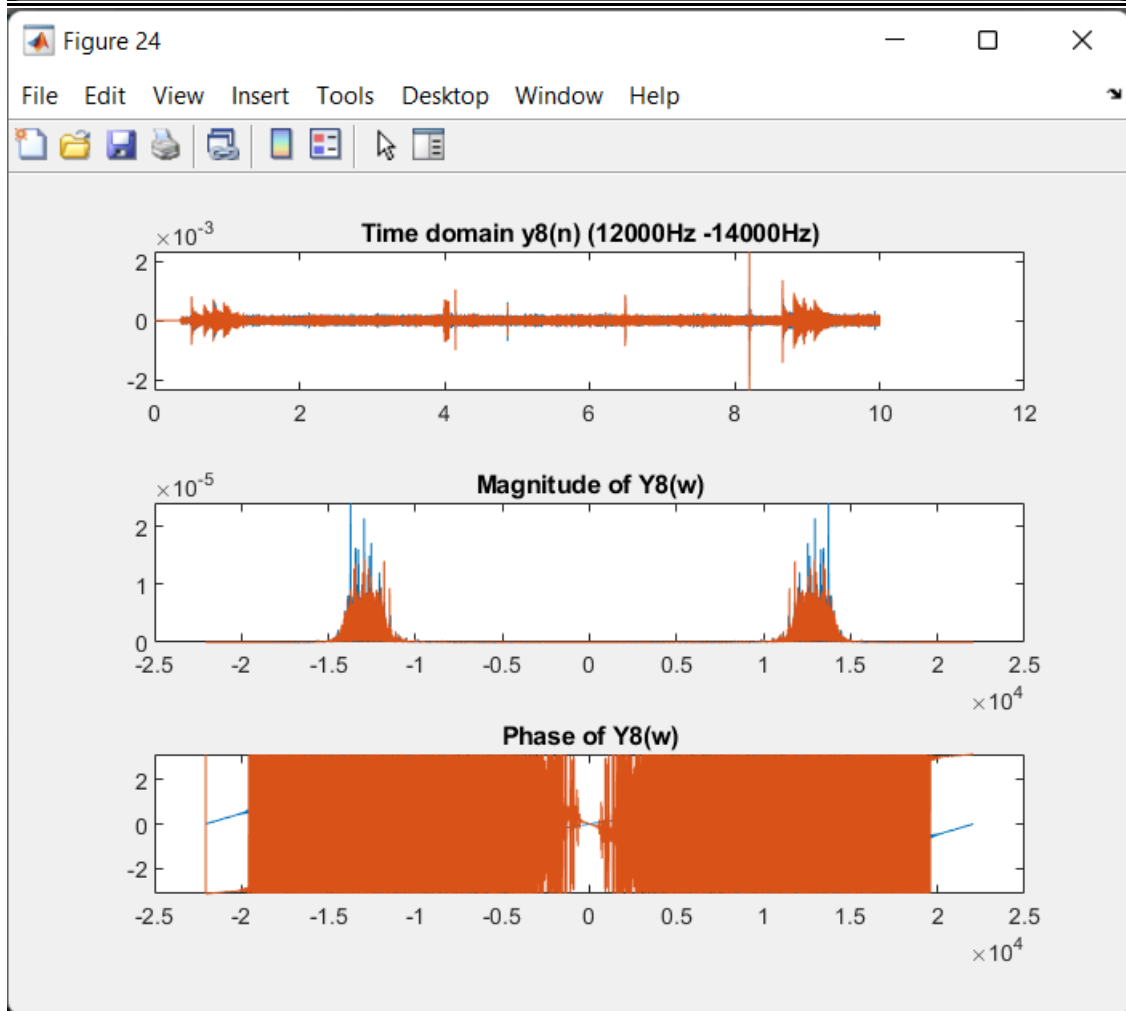
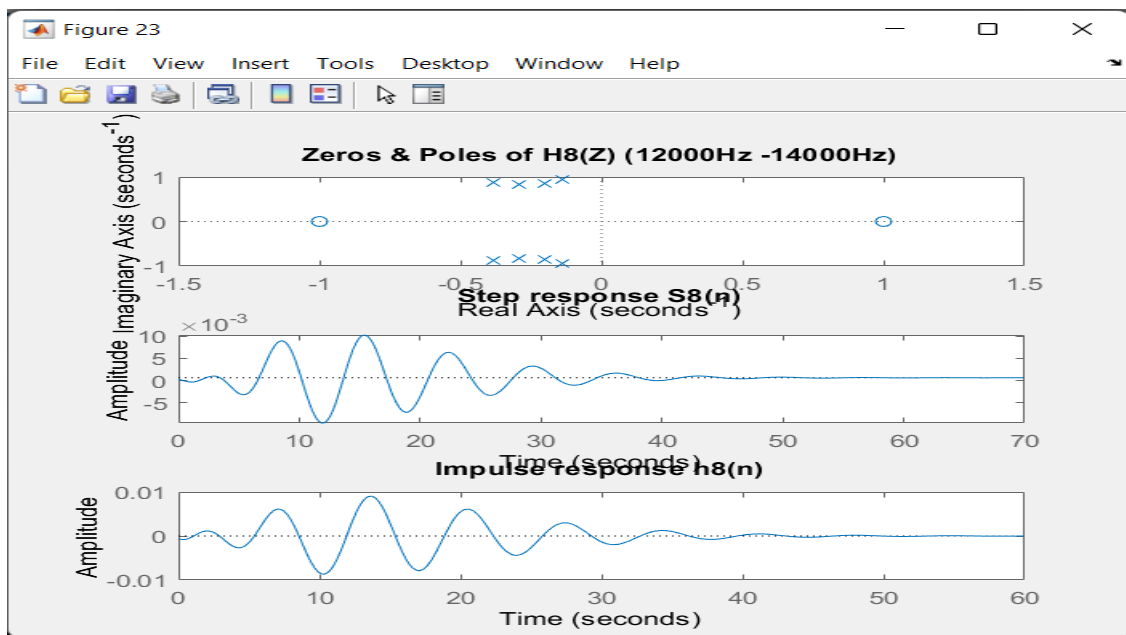


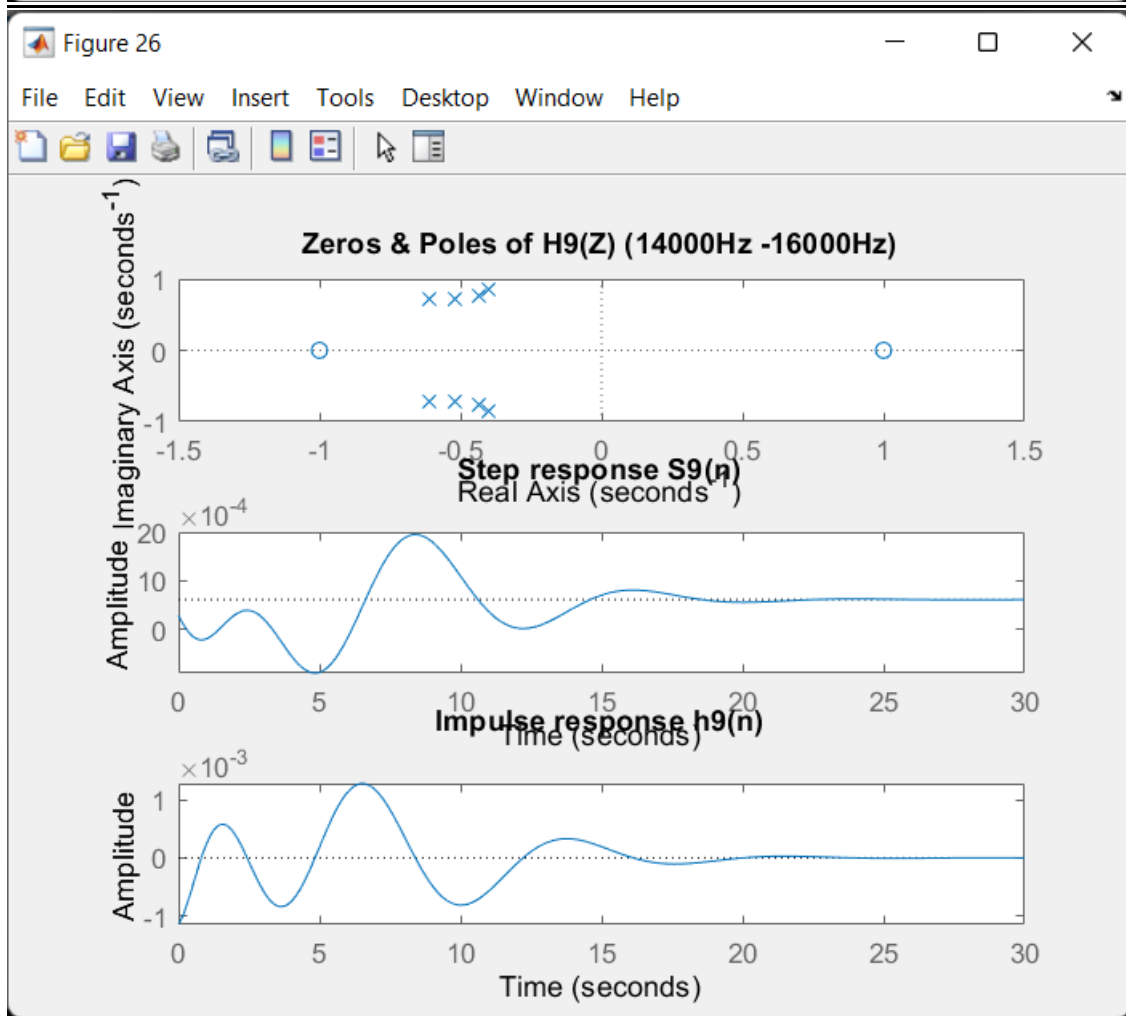
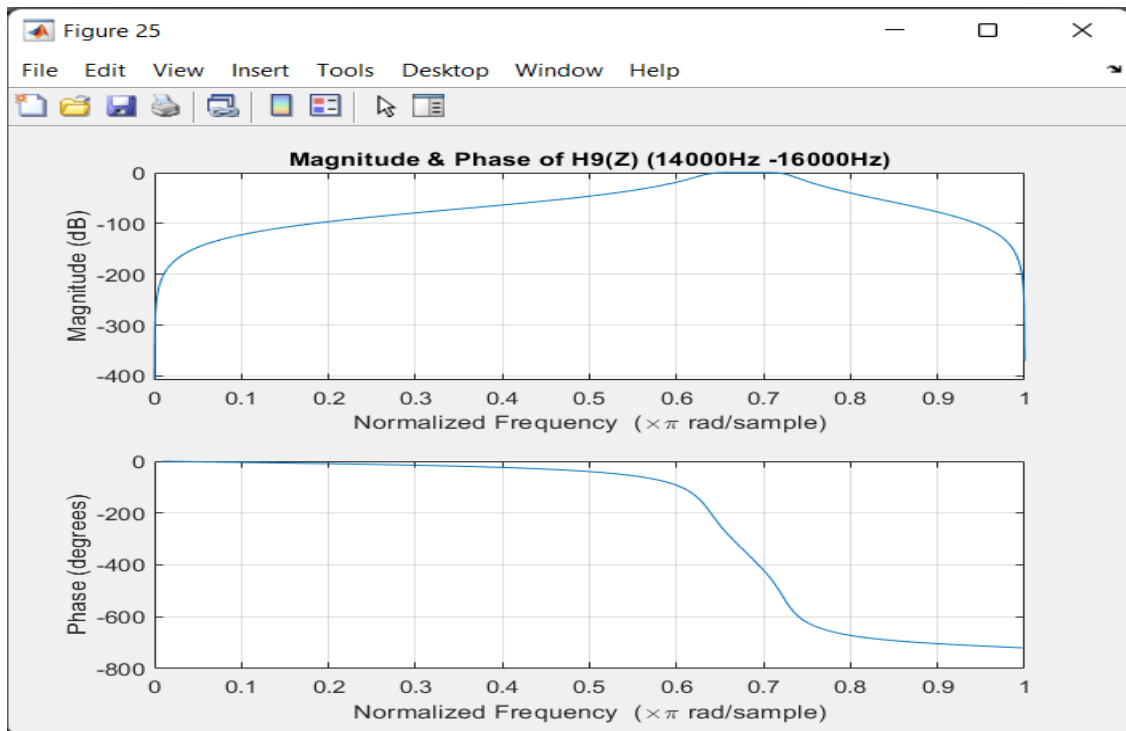


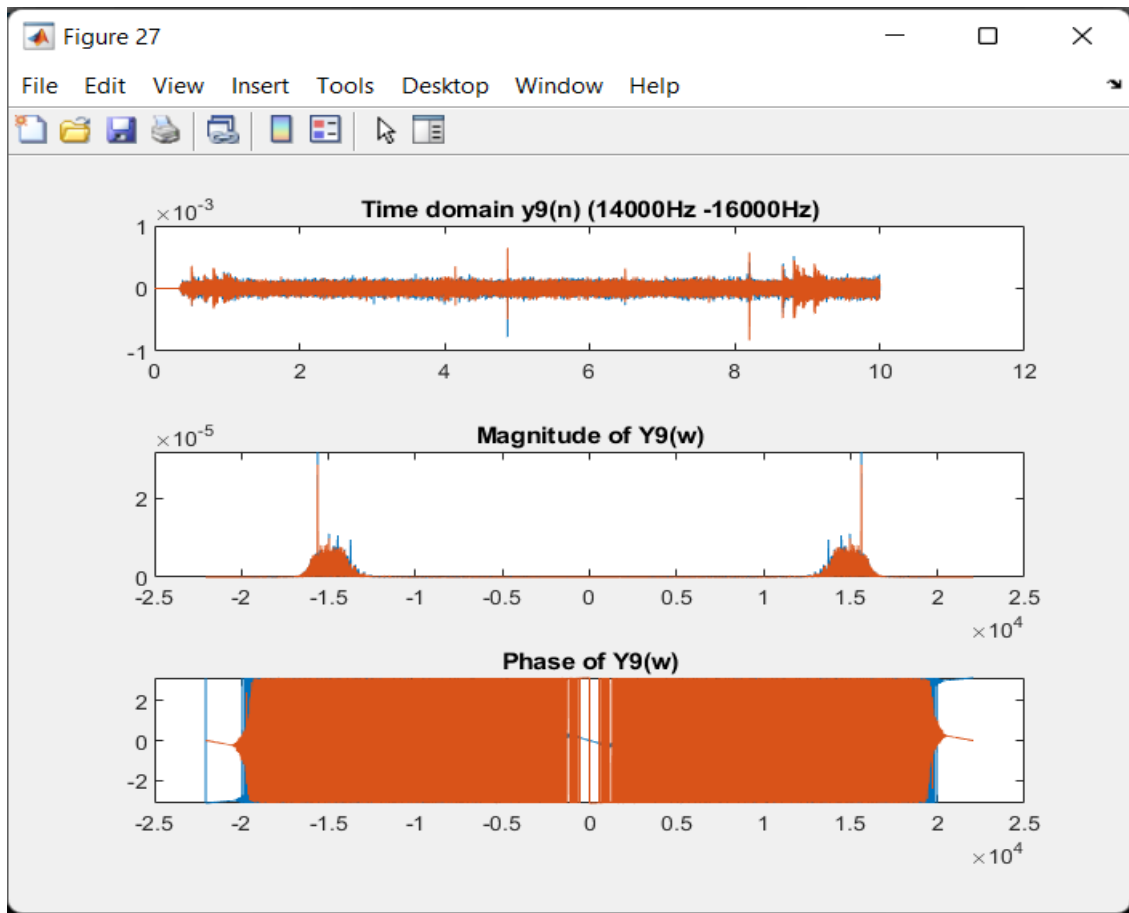




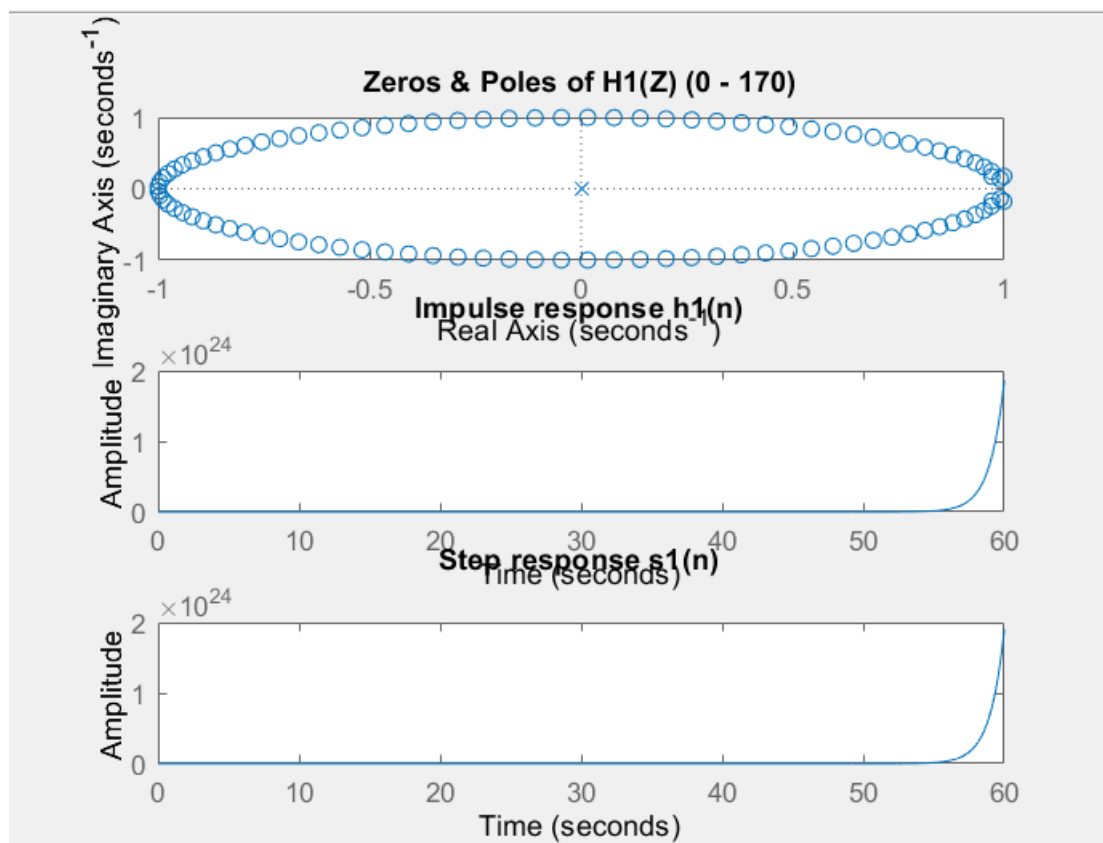
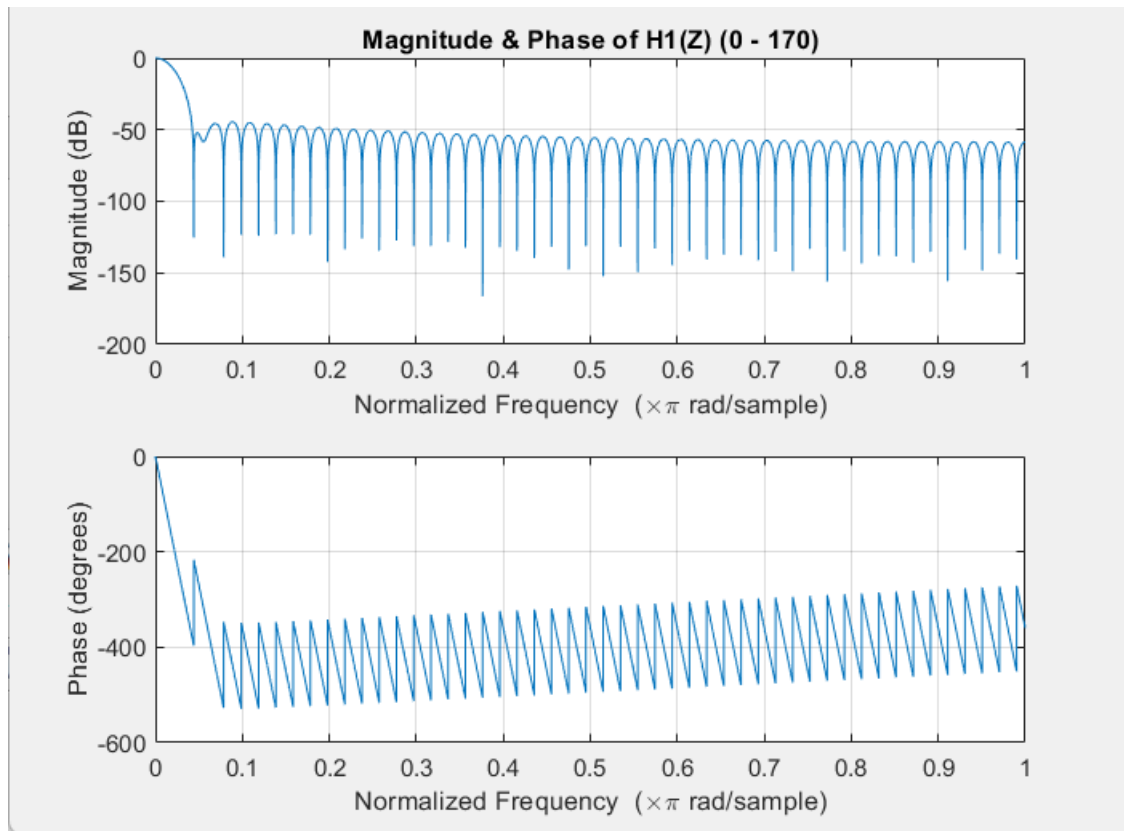


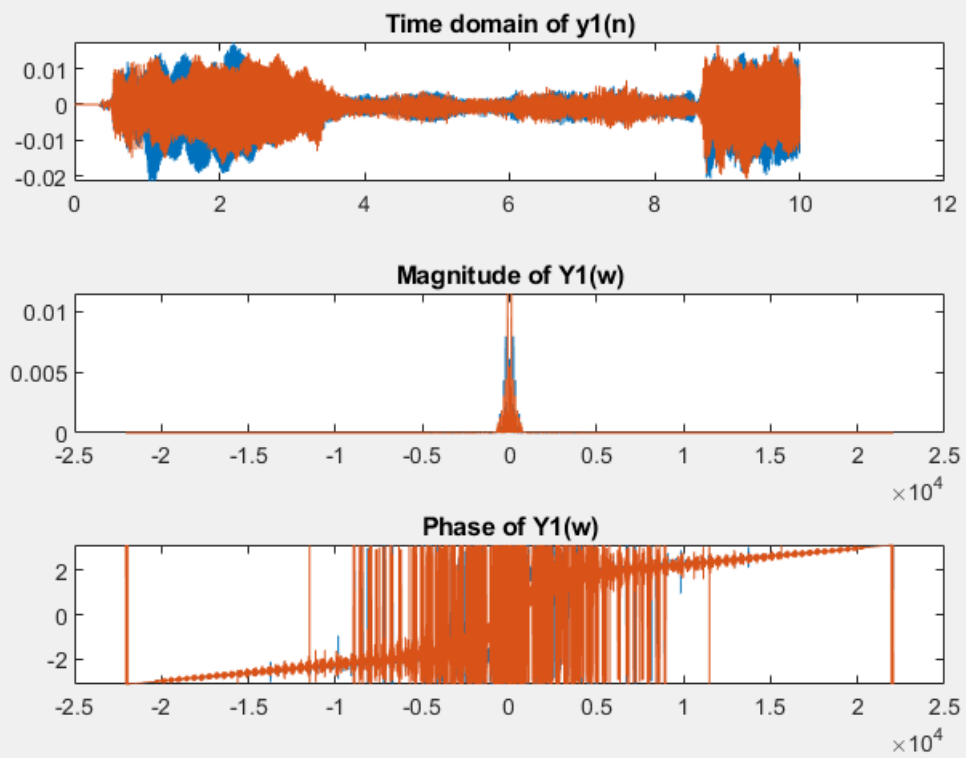






Doubling the Sample Rate





Decreasing to Half

